aws

Guida per gli sviluppatori

Amazon Managed Streaming per Apache Kafka

Amazon Managed Streaming per Apache Kafka: Guida per gli sviluppatori

Table of Contents

Benvenuto nella Guida per gli sviluppatori di Amazon MSK

Benvenuto nella Amazon Managed Streaming for Apache Kafka Developer Guide. I seguenti argomenti possono aiutarti a iniziare a usare questa guida, in base a ciò che stai cercando di fare.

- Crea un cluster MSK Provisioned seguendo il tutorial. Inizia a usare Amazon MSK
- Approfondisci le funzionalità di MSK Provisioned in. Che cos'è MSK Provisioned?
- Esegui Apache Kafka senza dover gestire e scalare la capacità del cluster con MSK Serverless.
- Utilizzate MSK Connect per lo streaming di dati da e verso il cluster Apache Kafka.
- Utilizzate MSK Replicator per replicare in modo affidabile i dati tra cluster MSK Provisioned diversi o uguali. Regioni AWS

Per i dettagli, le funzionalità principali e i prezzi del prodotto, consulta la pagina del servizio Amazon MSK.

Che cos'è Amazon MSK?

Amazon Managed Streaming for Apache Kafka (Amazon MSK) è un servizio completamente gestito che consente di costruire ed eseguire applicazioni che utilizzano Apache Kafka per elaborare i dati in streaming. Amazon MSK fornisce le operazioni del piano di controllo, ad esempio quelle per la creazione, l'aggiornamento e l'eliminazione di cluster. Consente di utilizzare operazioni del piano dati Apache Kafka, come quelle per la produzione e il consumo di dati. Esegue versioni open-source di Apache Kafka. Ciò significa che le applicazioni, gli strumenti e i plugin esistenti dei partner e della comunità Apache Kafka sono supportati senza richiedere modifiche al codice dell'applicazione. Puoi utilizzare Amazon MSK per creare cluster che utilizzano le versioni di Apache Kafka elencate nella sezione the section called “Versioni di Apache Kafka supportate”.

Questi componenti descrivono l'architettura di Amazon MSK:

- Nodi broker: quando crei un cluster Amazon MSK, specifichi quanti nodi broker desideri che Amazon MSK crei in ogni zona di disponibilità. Il minimo è un broker per zona di disponibilità. Ogni zona di disponibilità dispone di una propria sottorete VPC.

  Amazon MSK Provisioned offre due tipi di broker: Broker Amazon MSK Standard e. Broker Amazon MSK Express In MSK Serverless, MSK gestisce i nodi broker utilizzati per gestire il traffico e fornisce le risorse del server Kafka solo a livello di cluster.

- ZooKeeper nodi: Amazon MSK crea anche i ZooKeeper nodi Apache per te. Apache ZooKeeper è un server open source che consente un coordinamento distribuito altamente affidabile.
- KRaft controller: la community Apache Kafka è stata sviluppata KRaft per sostituire Apache per la gestione dei metadati nei cluster Apache ZooKeeper Kafka. In KRaft modalità, i metadati del cluster vengono propagati all'interno di un gruppo di controller Kafka, che fanno parte del cluster Kafka, anziché tra i nodi. ZooKeeper KRafti controller sono inclusi senza costi aggiuntivi per l'utente e non richiedono alcuna configurazione o gestione aggiuntiva da parte dell'utente.
- Produttori, consumatori e creatori di argomenti: Amazon MSK ti consente di utilizzare le operazioni sul piano dati di Apache Kafka per creare argomenti e produrre e utilizzare dati.
- Operazioni del cluster È possibile utilizzare AWS Management Console, the AWS Command Line Interface (AWS CLI) o the APIs nell'SDK per eseguire operazioni sul piano di controllo. Ad esempio, puoi creare o eliminare un cluster Amazon MSK, elencare tutti i cluster di un account, visualizzare le proprietà di un cluster e aggiornare il numero e il tipo di broker in un cluster.

Amazon MSK riconosce gli scenari di errore più comuni e avvia automaticamente il ripristino per cluster per permettere alle applicazioni produttore e consumatore di continuare le operazioni di scrittura e lettura con impatto minimo. Quando Amazon MSK rileva un errore del broker, attenua l'errore o sostituisce il broker non integro o non raggiungibile con uno nuovo. Inoltre, ove possibile, riutilizza lo storage del broker precedente per ridurre i dati che devono essere replicati da Apache Kafka. L'impatto sulla disponibilità è limitato al tempo richiesto da Amazon MSK per completare il rilevamento e il ripristino. Dopo un ripristino, le applicazioni produttore e consumatore possono continuare a comunicare con gli stessi indirizzi IP del broker utilizzati prima dell'errore.

Configurazione di Amazon MSK

Prima di utilizzare Amazon MSK per la prima volta, è necessario completare le seguenti operazioni.

Attività

- Iscriviti per AWS
- Esecuzione del download di librerie e strumenti

Iscriviti per AWS

Quando ti registri AWS, il tuo account Amazon Web Services viene automaticamente registrato per tutti i servizi in AWS, incluso Amazon MSK. Ti vengono addebitati solo i servizi che utilizzi.

Se hai già un AWS account, passa all'attività successiva. Se non disponi di un account AWS , utilizza la seguente procedura per crearne uno.

Registrazione per un account Amazon Web Services

1. Apri la https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

   Parte della procedura di registrazione prevede la ricezione di una telefonata o di un messaggio di testo e l'immissione di un codice di verifica sulla tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

Esecuzione del download di librerie e strumenti

Le librerie e gli strumenti seguenti semplificano l'utilizzo di Amazon MSK:

- L'AWS Command Line Interface (AWS CLI) supporta Amazon MSK. Ti AWS CLI consente di controllare più Amazon Web Services dalla riga di comando e di automatizzarli tramite script. Effettua AWS CLI l'upgrade alla versione più recente per assicurarti che supporti le funzionalità di Amazon MSK documentate in questa guida per l'utente. Per istruzioni dettagliate su come

aggiornare la AWS CLI, consulta la pagina Installing the AWS Command Line Interface. Dopo aver installato AWS CLI, devi configurarlo. Per informazioni su come configurare AWS CLI, consulta aws configure.

- La Documentazione di riferimento a Streaming gestito da Amazon per Apache Kafka documenta le operazioni dell'API supportate da Amazon MSK.
- Amazon Web Services SDKs for Go, Java, .NET JavaScript, Node.js, PHP, Python e Ruby includono il supporto e gli esempi di Amazon MSK.

Che cos'è MSK Provisioned?

I cluster Amazon MSK Provisioned offrono un'ampia gamma di caratteristiche e funzionalità per aiutarti a ottimizzare le prestazioni del cluster e soddisfare le tue esigenze di streaming. Gli argomenti seguenti descrivono la funzionalità in dettaglio.

MSK Provisioned è un'opzione di implementazione dei cluster MSK che consente di configurare e scalare manualmente i cluster Apache Kafka. Ciò offre diversi livelli di controllo sull'infrastruttura che alimenta l'ambiente Apache Kafka. Con MSK Provisioned, è possibile scegliere i tipi di istanze, i volumi di storage (broker standard) e il numero di nodi broker che costituiscono i cluster Kafka. È inoltre possibile scalare il cluster aggiungendo o rimuovendo broker man mano che le esigenze di elaborazione dei dati si evolvono. Questa flessibilità consente di ottimizzare i cluster per i requisiti specifici del carico di lavoro, sia che si tratti di massimizzare il throughput, la capacità di conservazione o altre caratteristiche prestazionali. Oltre alle opzioni di configurazione dell'infrastruttura, MSK Provisioned offre vantaggi operativi, di monitoraggio e di sicurezza di livello aziendale. Ciò include funzionalità come gli aggiornamenti della versione di Apache Kafka, la sicurezza integrata tramite crittografia e controllo degli accessi e l'integrazione con altre Servizi AWS come Amazon per il monitoraggio. CloudWatch MSK Provisioned offre due tipi principali di broker: Standard ed Express.

Per informazioni sull'API MSK Provisioned, consulta l'Amazon MSK API Reference.

Inizia a usare Amazon MSK

In questa sezione viene illustrato un esempio di come creare un cluster MSK, produrre e utilizzare dati, nonché monitorare l'integrità del cluster utilizzando i parametri. Questo esempio non rappresenta tutte le opzioni che è possibile scegliere quando si crea un cluster MSK. In diverse parti di questo tutorial verranno scelte opzioni predefinite per semplicità. Ciò non significa che siano le uniche opzioni che funzionano per la configurazione di un cluster MSK o delle istanze client.

Argomenti

Fase 1: creare un cluster MSK Provisioned

In questa fase di Guida introduttiva all'uso di Amazon MSK, crei un cluster Amazon MSK Provisioned. Per creare questo cluster, si utilizza l'opzione Quick create AWS Management Console in.

Per creare un cluster Amazon MSK utilizzando AWS Management Console

1. Accedere a e aprire la console Amazon MSK da https://console.aws.amazon.com/msk/casa? AWS Management Console region=us-east-1#/home/.
2. Scegli Create cluster (Crea cluster).
3. Per Metodo di creazione, lascia selezionata l'opzione Creazione rapida. L'opzione Creazione rapida consente di creare un cluster con le impostazioni predefinite.
4. Per Nome cluster, inserisci un nome descrittivo per il cluster. Ad esempio, **`MSKTutorialCluster`**.
5. Per le proprietà generali del cluster, procedi come segue:
   a. Per il tipo di cluster, scegli Provisioned.
   b. Scegli una versione di Apache Kafka da eseguire sui broker. Scegli Visualizza compatibilità tra le versioni per visualizzare una tabella di confronto.
   c. Per il tipo di broker, scegli i broker Standard o Express.
   d. Scegli la dimensione del broker.
6. Dalla tabella in Tutte le impostazioni del cluster, copia i valori delle seguenti impostazioni e salvali perché ti serviranno più avanti in questo tutorial:
   - VPC
   - Sottoreti
   - Gruppi di sicurezza associati al VPC
7. Scegli Create cluster (Crea cluster).
8. Verifica lo Stato del cluster nella pagina Riepilogo del cluster. Quando Amazon MSK assegna il cluster, lo stato passa da Creazione in corso ad Attivo. Quando lo stato è Attivo, puoi connetterti

al cluster. Per ulteriori informazioni sugli stati del cluster, consulta la pagina Comprendi gli stati del cluster MSK Provisioned.

Fase successiva

Fase 2: creazione di un ruolo IAM che conceda l'accesso alla creazione di argomenti nel cluster Amazon MSK

Fase 2: creazione di un ruolo IAM che conceda l'accesso alla creazione di argomenti nel cluster Amazon MSK

In questo passaggio, eseguirai due attività. La prima attività consiste nel creare una policy IAM che consenta l'accesso alla creazione di argomenti nel cluster e all'invio di dati a tali argomenti. La seconda attività consiste nel creare un ruolo IAM e associarvi questa policy. In un passaggio successivo, si crea un computer client che assume questo ruolo e lo utilizza per creare un argomento nel cluster e per inviare dati a quell'argomento.

Creazione di una policy IAM che consenta di creare argomenti e scrivere su di essi

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.
3. Scegliere Create Policy (Crea policy).
4. In Policy editor, scegli JSON, quindi sostituisci il JSON nella finestra dell'editor con il seguente JSON.

   Nell'esempio seguente, sostituisci quanto segue:

   - *region*con il codice del Regione AWS luogo in cui hai creato il cluster.
   - Esempio di ID dell'account*123456789012*, con il tuo Account AWS ID.
   - *MSKTutorialCluster*e*MSKTutorialCluster*/*7d7131e1-25c5-4e9a-9ac5-ea85bee4da11-14*, con il nome del cluster e il relativo ID.

   JSON

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

```
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:AlterCluster",
                "kafka-cluster:DescribeCluster"
            ],
            "Resource": [
                "arn:aws:kafka:us-
east-1:123456789012:cluster/MSKTutorialCluster/7d7131e1-25c5-4e9a-9ac5-
ea85bee4da11-14"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:*Topic*",
                "kafka-cluster:WriteData",
                "kafka-cluster:ReadData"
            ],
            "Resource": [
            "arn:aws:kafka:us-east-1:123456789012:topic/MSKTutorialCluster/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeGroup"
            ],
            "Resource": [
            "arn:aws:kafka:us-east-1:123456789012:group/MSKTutorialCluster/*"
            ]
        }
    ]
}
```

Per istruzioni su come scrivere policy sicure, consultathe section called “Controllo degli accessi IAM”.

5. Scegli Next (Successivo).
6. Nella pagina Rivedi e crea, effettua le operazioni seguenti:

a. Per Nome della politica, inserisci un nome descrittivo, ad esempio**`msk-tutorial-policy`**.

b. In Autorizzazioni definite in questa politica, rivedi e and/or modifica le autorizzazioni definite nella tua politica.

c. (Facoltativo) Per facilitare l'identificazione, l'organizzazione o la ricerca della politica, scegli Aggiungi nuovo tag per aggiungere tag come coppie chiave-valore. Ad esempio, aggiungi un tag alla tua politica con la coppia chiave-valore e. **`Environment Test`**

   Per ulteriori informazioni sull'utilizzo dei tag, consulta Tags for AWS Identity and Access Management resources nella IAM User Guide.

7. Scegliere Create Policy (Crea policy).

Creazione di un ruolo IAM e collegamento della policy al ruolo

1. Nel riquadro di navigazione, scegli Ruoli, quindi scegli Crea ruolo.
2. Nella pagina Seleziona un'entità attendibile, esegui le operazioni seguenti:

   a. Per Trusted entity type (Tipo di entità attendibile), scegli Servizio AWS.

   b. Per Servizio o caso d'uso, scegli EC2.

   c. In Use case (Caso d'uso), scegli EC2.

3. Scegli Next (Successivo).
4. Nella pagina Add permissions (Aggiungi autorizzazioni), esegui le operazioni seguenti:

   a. Nella casella di ricerca sotto Politiche di autorizzazione, inserisci il nome della politica che hai creato in precedenza per questo tutorial. Quindi, scegli la casella a sinistra del nome della politica.

   b. (Facoltativo) Impostare un limite delle autorizzazioni. Questa è una caratteristica avanzata disponibile per i ruoli di servizio, ma non per i ruoli collegati ai servizi. Per informazioni sull'impostazione di un limite di autorizzazioni, consulta Creating roles and attaching policies (console) nella IAM User Guide.

5. Scegli Next (Successivo).
6. Nella pagina Name, review, and create (Assegna un nome, rivedi e crea), esegui le operazioni seguenti:

   a. Per il nome del ruolo, inserisci un nome descrittivo, ad esempio. **`msk-tutorial-role`**

> **Important**
>
> Quando assegni un nome a un ruolo, tieni presente quanto segue:
>
> - I nomi dei ruoli devono essere univoci all'interno del tuo Account AWS account e non possono essere resi unici per caso.
>
> Ad esempio, non creare ruoli denominati **`PRODROLE`** e **`prodrole`**. Quando il nome di un ruolo viene utilizzato in una policy o come parte di un ARN, il nome del ruolo fa distinzione tra maiuscole e minuscole, tuttavia quando un nome di ruolo viene visualizzato ai clienti nella console, ad esempio durante il processo di accesso, il nome del ruolo non fa distinzione tra maiuscole e minuscole.
> - Non è possibile modificare il nome del ruolo dopo averlo creato, in quanto altre entità possono fare riferimento al ruolo.

b. (Facoltativo) In Descrizione, inserisci una descrizione per il ruolo.

c. (Facoltativo) Per modificare i casi d'uso e le autorizzazioni per il ruolo, nel Passaggio 1: Seleziona le entità attendibili o nel Passaggio 2: Aggiungi le sezioni relative alle autorizzazioni, scegli Modifica.

d. (Facoltativo) Per facilitare l'identificazione, l'organizzazione o la ricerca del ruolo, scegli Aggiungi nuovo tag per aggiungere tag come coppie chiave-valore. Ad esempio, aggiungi un tag al tuo ruolo con la coppia chiave-valore e. **`ProductManager John`**

   Per ulteriori informazioni sull'utilizzo dei tag, consulta Tags for AWS Identity and Access Management resources nella IAM User Guide.

7. Verificare il ruolo e quindi scegliere Create role (Crea ruolo).

Fase successiva

Passaggio 3: creazione di un computer client

Passaggio 3: creazione di un computer client

In questa fase di Guida introduttiva all'uso di Amazon MSK, crei una macchina client. Utilizza questo computer client per creare un argomento che produce e consuma dati. Per semplicità, creerai questo computer client nel VPC associato al cluster MSK in modo che il client possa connettersi facilmente al cluster.

Per creare un computer client

1. Apri la EC2 console Amazon all'indirizzo https://console.aws.amazon.com/ec2/.
2. Dalla dashboard della EC2 console Amazon, scegli Launch instance.
3. In Nome e tag, in Nome, inserisci un nome descrittivo per la tua macchina client in modo da poterne tenere traccia facilmente. Ad esempio, **`MSKTutorialClient`**.
4. In Immagini di applicazioni e sistemi operativi (Amazon Machine Image), per Amazon Machine Image (AMI), scegli Amazon Linux 2 AMI (HVM) - Kernel 5.10, tipo di volume SSD.
5. Ad esempio, mantieni la selezione predefinita di t2.micro.
6. In Coppia di chiavi (login), scegli una coppia di chiavi esistente o creane una nuova. Se non hai bisogno di una coppia di chiavi per connetterti alla tua istanza, puoi scegliere Procedi senza una coppia di chiavi (scelta non consigliata).

   Per creare una nuova coppia di chiavi, eseguire quanto descritto di seguito:

   a. Scegli Crea una nuova coppia di chiavi.
   b. Per Key pair name (Nome coppia di chiavi), inserire **`MSKKeyPair`**.
   c. Per Tipo di coppia di chiavi e Formato di file con chiave privata, mantieni le selezioni predefinite.
   d. Scegliere Create key pair (Crea coppia di chiavi).

   In alternativa, è possibile utilizzare una coppia di chiavi esistente.
7. Scorri la pagina verso il basso ed espandi la sezione Dettagli avanzati, quindi procedi come segue:

   - Per il profilo dell'istanza IAM, scegli un ruolo IAM che desideri venga assunto dalla macchina client.

     Se non disponi di un ruolo IAM, procedi come segue:

     i. Scegli Crea nuovo profilo IAM.
     ii. Esegui i passaggi indicati nella Fase 2: Crea un ruolo IAM.
8. Scegliere Launch Instance (Avvia istanza).
9. Scegliere View Instances (Vedi istanze). Quindi, nella colonna Gruppi di sicurezza, scegli il gruppo di sicurezza associato alla nuova istanza. Copia l'ID del gruppo di sicurezza e salvalo per un secondo momento.

10. Apri la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
11. Fai clic su Security Groups (Gruppi di sicurezza) nel pannello di navigazione. Trova il gruppo di sicurezza del quale hai salvato l'ID in the section called “Creazione di un cluster”.
12. Nella scheda Regole in entrata, scegli Modifica le regole in entrata.
13. Scegli Aggiungi regola.
14. Nella nuova regola, scegliere All traffic (Tutto il traffico) nella colonna Type (Tipo) . Nel secondo campo della colonna Origine, inserisci l'ID del gruppo di sicurezza del computer client. Questo è il gruppo il cui nome hai salvato dopo aver avviato l'istanza del computer client.
15. Scegliere Salva regole. Ora il gruppo di sicurezza del cluster può accettare il traffico proveniente dal gruppo di sicurezza del computer client.

Fase successiva

Fase 4: creare un argomento nel cluster Amazon MSK

Fase 4: creare un argomento nel cluster Amazon MSK

In questo passaggio della Guida introduttiva all'utilizzo di Amazon MSK, installi librerie e strumenti client di Apache Kafka sul computer client e quindi crei un argomento.

> **Warning**
>
> I numeri di versione di Apache Kafka utilizzati in questo tutorial sono solo degli esempi. Ti consigliamo di utilizzare la stessa versione del client della versione del cluster MSK. In una versione precedente del client potrebbero mancare alcune funzionalità e correzioni di bug critici.

Argomenti

- Determinazione della versione del cluster MSK
- Creazione di un argomento sul computer client

Determinazione della versione del cluster MSK

1. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/.
2. Nella barra di navigazione, scegli la regione in cui hai creato il cluster MSK.

3. Scegliete il cluster MSK.
4. Prendi nota della versione di Apache Kafka utilizzata nel cluster.
5. Sostituisci le istanze dei numeri di versione di Amazon MSK in questo tutorial con la versione ottenuta nel passaggio 3.

Creazione di un argomento sul computer client

1. Connect al computer client.
   a. Apri la EC2 console Amazon all'indirizzo https://console.aws.amazon.com/ec2/.
   b. Nel riquadro di navigazione, scegliere Instances (Istanze). Quindi, seleziona la casella di controllo accanto al nome del computer client in cui hai creatoPassaggio 3: creazione di un computer client.
   c. Scegliere Actions (Operazioni), quindi selezionare Connect (Connetti). Segui le istruzioni riportate nella console per connetterti al computer client.
2. Installa Java e configura la variabile di ambiente della versione Kafka.
   a. Installa Java sul computer client eseguendo il comando seguente.

```
sudo yum -y install java-11
```

   b. Memorizza la versione Kafka del tuo cluster MSK nella variabile di ambienteKAFKA_VERSION, come mostrato nel comando seguente. Avrai bisogno di queste informazioni durante tutta la configurazione.

```
export KAFKA_VERSION={KAFKA VERSION}
```

   Ad esempio, se utilizzi la versione 3.6.0, usa il comando seguente.

```
export KAFKA_VERSION=3.6.0
```

3. Scarica ed estrai Apache Kafka.
   a. Eseguire il seguente comando per scaricare Apache Kafka.

```
wget https://archive.apache.org/dist/kafka/$KAFKA_VERSION/kafka_2.13-
$KAFKA_VERSION.tgz
```

> **Note**
>
> L'elenco seguente presenta alcune informazioni alternative per il download di Kafka che è possibile utilizzare in caso di problemi.
>
> - Se riscontri problemi di connettività o desideri utilizzare un sito mirror, prova a utilizzare il selettore dei mirror Apache, come mostrato nel comando seguente.
>
>   ```
>   wget https://www.apache.org/dyn/closer.cgi?path=/kafka/$KAFKA_VERSION/kafka_2.13-$KAFKA_VERSION.tgz
>   ```
>
> - Scaricate una versione appropriata direttamente dal sito Web di Apache Kafka.

b. Eseguire il comando seguente nella directory in cui è stato scaricato il file TAR nella fase precedente.

```
tar -xzf kafka_2.13-$KAFKA_VERSION.tgz
```

c. Memorizza il percorso completo della directory appena creata all'interno della `KAFKA_ROOT` variabile di ambiente.

```
export KAFKA_ROOT=$(pwd)/kafka_2.13-$KAFKA_VERSION
```

4. Configura l'autenticazione per il tuo cluster MSK.

a. Trova la versione più recente della libreria client Amazon MSK IAM. Questa libreria consente alla macchina client di accedere al cluster MSK utilizzando l'autenticazione IAM.

b. Utilizzando i seguenti comandi, accedi alla `$KAFKA_ROOT/libs` directory e scarica il JAR Amazon MSK IAM associato che hai trovato nel passaggio precedente. Assicurati di sostituirlo *`{LATEST VERSION}`* con il numero di versione effettivo che stai scaricando.

```
cd $KAFKA_ROOT/libs
```

```
wget https://github.com/aws/aws-msk-iam-auth/releases/latest/download/aws-msk-iam-auth-{LATEST VERSION}-all.jar
```

> **Note**
>
> Prima di eseguire qualsiasi comando Kafka che interagisca con il tuo cluster MSK, potresti dover aggiungere il file JAR Amazon MSK IAM al tuo classpath Java. Imposta la variabile di `CLASSPATH` ambiente, come mostrato nell'esempio seguente.
>
> ```
> export CLASSPATH=$KAFKA_ROOT/libs/aws-msk-iam-auth-{LATEST VERSION}-
> all.jar
> ```
>
> Questo imposta la `CLASSPATH` per l'intera sessione, rendendo il JAR disponibile per tutti i comandi Kafka successivi.

c. Vai alla `$KAFKA_ROOT/config` directory per creare il file di configurazione del client.

```
cd $KAFKA_ROOT/config
```

d. Copia le impostazioni delle proprietà seguenti e incollale in un nuovo file. Salva il file con nome **`client.properties`**.

```
security.protocol=SASL_SSL
sasl.mechanism=AWS_MSK_IAM
sasl.jaas.config=software.amazon.msk.auth.iam.IAMLoginModule required;
sasl.client.callback.handler.class=software.amazon.msk.auth.iam.IAMClientCallbackHandle
```

5. (Facoltativo) Regola la dimensione dell'heap Java per gli strumenti Kafka.

   Se riscontrate problemi relativi alla memoria o state lavorando con un gran numero di argomenti o partizioni, potete modificare la dimensione dell'heap Java. Per fare ciò, imposta la variabile di `KAFKA_HEAP_OPTS` ambiente prima di eseguire i comandi di Kafka.

   L'esempio seguente imposta la dimensione massima e iniziale dell'heap su 512 megabyte. Regola questi valori in base ai requisiti specifici e alle risorse di sistema disponibili.

   ```
   export KAFKA_HEAP_OPTS="-Xmx512M -Xms512M"
   ```

6. Ottieni le informazioni sulla connessione al cluster.

   a. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/.

b. Attendi che lo stato del cluster diventi Attivo. Questo processo potrebbe richiedere diversi minuti. Dopo che lo stato diventa Attivo, scegli il nome del cluster. Verrà visualizzata una pagina contenente il riepilogo del cluster.

c. Scegli Visualizza le informazioni sul client.

d. Copia la stringa di connessione per l'endpoint privato.

   Otterrai tre endpoint per ciascuno dei broker. Memorizza una di queste stringhe di connessione nella variabile di ambienteBOOTSTRAP_SERVER, come illustrato nel comando seguente. Sostituire *<bootstrap-server-string>* con il valore effettivo della stringa di connessione.

```
export BOOTSTRAP_SERVER=<bootstrap-server-string>
```

7. Eseguite il comando seguente per creare l'argomento.

```
$KAFKA_ROOT/bin/kafka-topics.sh --create --bootstrap-server $BOOTSTRAP_SERVER
 --command-config $KAFKA_ROOT/config/client.properties --replication-factor 3 --
partitions 1 --topic MSKTutorialTopic
```

Se ottenete un `NoSuchFileException` `client.properties` file, assicuratevi che questo file esista nella directory di lavoro corrente all'interno della cartella bin di Kafka.

> **Note**
>
> Se preferite non impostare la variabile di `CLASSPATH` ambiente per l'intera sessione, potete in alternativa aggiungere la variabile come prefisso a ogni comando di Kafka. `CLASSPATH` Questo approccio applica il classpath solo a quel comando specifico.
>
> ```
> CLASSPATH=$KAFKA_ROOT/libs/aws-msk-iam-auth-{LATEST VERSION}-all.jar \
> $KAFKA_ROOT/bin/kafka-topics.sh --create \
> --bootstrap-server $BOOTSTRAP_SERVER \
> --command-config $KAFKA_ROOT/config/client.properties \
> --replication-factor 3 \
> --partitions 1 \
> --topic MSKTutorialTopic
> ```

8. (Facoltativo) Verificate che l'argomento sia stato creato correttamente.

a. Se il comando ha esito positivo, dovrebbe apparire il seguente messaggio: `Created topic MSKTutorialTopic.`

b. Elenca tutti gli argomenti per confermare l'esistenza dell'argomento.

```
$KAFKA_ROOT/bin/kafka-topics.sh --list --bootstrap-server $BOOTSTRAP_SERVER --
command-config $KAFKA_ROOT/config/client.properties
```

Se il comando ha esito negativo o si verifica un errore, consulta Risolvi i problemi del tuo cluster Amazon MSK per informazioni sulla risoluzione dei problemi.

9. (Facoltativo) Eliminate le variabili di ambiente utilizzate in questo tutorial.

   Se desideri mantenere le variabili di ambiente per i passaggi successivi di questo tutorial, salta questo passaggio. Altrimenti, potete annullare l'impostazione di queste variabili, come illustrato nell'esempio seguente.

```
unset KAFKA_VERSION KAFKA_ROOT BOOTSTRAP_SERVER CLASSPATH KAFKA_HEAP_OPTS
```

Fase successiva

Passaggio 5: produzione e utilizzo di dati

Passaggio 5: produzione e utilizzo di dati

In questa fase di Get Started Using Amazon MSK, produci e consumi dati.

Per produrre e consumare messaggi

1. Eseguire il comando seguente per avviare un produttore della console.

```
$KAFKA_ROOT/bin/kafka-console-producer.sh --broker-list $BOOTSTRAP_SERVER --
producer.config $KAFKA_ROOT/config/client.properties --topic MSKTutorialTopic
```

2. Immettere qualsiasi messaggio desiderato e premere Enter (Invio). Ripetere questa fase due o tre volte. Ogni volta che si immette una riga e si preme Enter (Invio), tale riga viene inviata al cluster Apache Kafka come un messaggio separato.
3. Mantenere aperta la connessione al computer client, quindi aprire una seconda connessione separata al computer in una nuova finestra. Poiché si tratta di una nuova sessione, imposta

nuovamente le variabili KAFKA_ROOT e di BOOTSTRAP_SERVER ambiente. Per informazioni su come impostare queste variabili di ambiente, consultaCreazione di un argomento sul computer client.

4. Esegui il comando seguente con la tua seconda stringa di connessione al computer client per creare un utente di console.

```
$KAFKA_ROOT/bin/kafka-console-consumer.sh --bootstrap-server $BOOTSTRAP_SERVER --
consumer.config $KAFKA_ROOT/config/client.properties --topic MSKTutorialTopic --
from-beginning
```

   Dovresti iniziare a vedere i messaggi che hai inserito in precedenza quando hai usato il comando console producer.

5. Immettere altri messaggi nella finestra del produttore e guardali apparire nella finestra del consumatore.

Fase successiva

Fase 6: Usa Amazon CloudWatch per visualizzare i parametri di Amazon MSK

Fase 6: Usa Amazon CloudWatch per visualizzare i parametri di Amazon MSK

In questa fase di Guida introduttiva all'uso di Amazon MSK, esamini i parametri di Amazon MSK in Amazon. CloudWatch

Per visualizzare i parametri di Amazon MSK in CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.
3. Scegliere la scheda All metrics (Tutti i parametri), quindi selezionare AWS/Kafka.
4. Per visualizzare i parametri a livello di broker, scegliere Broker ID, Cluster Name (ID broker, nome cluster). Per i parametri a livello di cluster, scegliere Cluster Name (Nome cluster).
5. (Facoltativo) Nel riquadro grafico, selezionate una statistica e un periodo di tempo, quindi create un CloudWatch allarme utilizzando queste impostazioni.

Fase successiva

Passaggio 7: Eliminare le AWS risorse create per questo tutorial

Passaggio 7: Eliminare le AWS risorse create per questo tutorial

Nel passaggio finale della Guida introduttiva all'utilizzo di Amazon MSK, elimini il cluster MSK e il computer client che hai creato per questo tutorial.

Per eliminare le risorse utilizzando il AWS Management Console

1. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/.
2. Scegli il nome del cluster. Ad esempio, MSKTutorialCluster.
3. Selezionare Actions (Operazioni), quindi selezionare Delete (Elimina).
4. Apri la EC2 console Amazon all'indirizzo https://console.aws.amazon.com/ec2/.
5. Scegli l'istanza che hai creato per il computer client, ad esempio **`MSKTutorialClient`**.
6. Scegli Stato istanza, quindi scegli Termina istanza.

Eliminazione del ruolo e della policy IAM

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Ruoli.
3. Nella casella di ricerca, inserisci il nome del ruolo IAM creato per questo tutorial.
4. Seleziona il ruolo. Quindi scegli Elimina ruolo e conferma l'eliminazione.
5. Nel riquadro di navigazione, seleziona Policy.
6. Nella casella di ricerca, inserisci il nome della policy creata per questo tutorial.
7. Scegli la policy per aprirne la pagina di riepilogo. Nella pagina di Riepilogo della policy, seleziona Elimina policy.
8. Scegli Delete (Elimina).

Amazon MSK: come funziona

Amazon MSK è un servizio Apache Kafka completamente gestito che semplifica la creazione e l'esecuzione di applicazioni che utilizzano Apache Kafka per elaborare dati in streaming. Questa guida fornisce informazioni per aiutare gli sviluppatori a capire come funziona Amazon MSK e come utilizzarlo efficacemente nelle loro applicazioni.

Ad alto livello, Amazon MSK fornisce un cluster Apache Kafka completamente gestito, fornito e gestito da. AWS Ciò significa che non devi preoccuparti del provisioning delle EC2 istanze, della configurazione delle impostazioni di rete, della gestione dei broker Kafka o dell'esecuzione di attività di manutenzione continue. Invece, puoi concentrarti sulla creazione della tua applicazione e lasciare che Amazon MSK gestisca l'infrastruttura. Amazon MSK effettua automaticamente il provisioning delle risorse di calcolo, storage e rete necessarie e fornisce funzionalità come scalabilità automatica, alta disponibilità e failover per garantire che il cluster Kafka sia affidabile e altamente disponibile. Questa guida illustra i componenti chiave di Amazon MSK e come utilizzarli per creare applicazioni di streaming di dati.

Gestisci il tuo cluster Provisioned

Un cluster Amazon MSK è la risorsa Amazon MSK primaria che puoi creare nel tuo account. Negli argomenti di questa sezione viene descritto come eseguire le operazioni comuni di Amazon MSK. Per un elenco di tutte le operazioni che puoi eseguire su un cluster MSK, consulta le risorse seguenti:

- Il AWS Management Console
- La documentazione di riferimento all'API di Amazon MSK
- La documentazione di riferimento ai comandi della CLI di Amazon MSK

Argomenti

- Contrassegna un tag a un cluster Amazon MSK
- Migrazione a un cluster Amazon MSK
- Eliminare un cluster Amazon MSK Provisioned

Crea un cluster MSK Provisioned

> **Important**
>
> Non è possibile modificare il VPC per un cluster MSK Provisioned dopo averlo creato.

Prima di poter creare un cluster MSK Provisioned, è necessario disporre di un (Amazon Virtual Private Cloud VPC) e configurare le sottoreti all'interno di tale VPC.

Per i broker Standard nella regione Stati Uniti occidentali (California settentrionale), sono necessarie due sottoreti in due zone di disponibilità diverse. Per altre regioni in cui è disponibile Amazon MSK, è possibile specificare due o tre sottoreti. Le sottoreti devono trovarsi tutte in zone di disponibilità differenti. Per i broker Express, sono necessarie tre sottoreti in tre diverse zone di disponibilità. Quando crei un cluster MSK Provisioned, Amazon MSK distribuisce i nodi broker in modo uniforme sulle sottoreti specificate.

Argomenti

- Crea un cluster MSK Provisioned utilizzando il AWS Management Console
- Crea un cluster Amazon MSK con provisioning utilizzando AWS CLI
- Crea un cluster MSK Provisioned con una configurazione Amazon MSK personalizzata utilizzando AWS CLI
- Crea un cluster MSK Provisioned utilizzando l'API Amazon MSK

Crea un cluster MSK Provisioned utilizzando il AWS Management Console

Le procedure riportate in questo argomento descrivono l'attività comune di creazione di un cluster MSK Provisioned utilizzando l'opzione di creazione personalizzata disponibile in. AWS Management Console Utilizzando le altre opzioni disponibili in AWS Management Console, è inoltre possibile creare quanto segue:

- Un cluster senza server

- Un cluster MSK Provisioned che utilizza l'opzione Quick create

Procedure in questo argomento

Fase 1: Configurazione e configurazione iniziali del cluster

1. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/.
2. Scegli Create cluster (Crea cluster).
3. Per il metodo di creazione del cluster, scegli Creazione personalizzata.
4. Per Nome cluster, specifica un nome univoco e contenente non più di 64 caratteri.
5. Per il tipo di cluster, scegli Provisioned.
6. Per la versione Apache Kafka, scegli una versione da eseguire sui broker. Per visualizzare un confronto tra le funzionalità di Amazon MSK supportate da ciascuna versione di Apache Kafka, scegli Visualizza compatibilità della versione.
7. Nella sezione Broker, procedi come segue:

    a. Per il tipo di broker, scegli una delle seguenti opzioni:

    - Broker Express: broker scalabili e ad alte prestazioni con storage virtuale completamente gestito. Scegli questo tipo di broker per applicazioni impegnative e ad alto rendimento.
    - Broker standard: broker Kafka tradizionale con controllo completo della configurazione. Scegli questo tipo di broker per carichi di lavoro generici con requisiti di throughput moderati.

    Per ulteriori informazioni su questi tipi di broker, consulta. Tipi di broker Amazon MSK

    b. Per Broker size, scegli una dimensione da utilizzare per il cluster in base alle esigenze di calcolo, memoria e archiviazione del cluster.

c. Per Numero di zone, scegli il numero di aree AWS Zone di disponibilitàin cui sono distribuiti i broker.

   I broker Express richiedono tre zone di disponibilità per una maggiore disponibilità.

d. Per i broker per zona, specifica il numero di broker che desideri che Amazon MSK crei in ogni zona di disponibilità. Il minimo è un broker per zona di disponibilità e il massimo è 30 broker per cluster per cluster ZooKeeper basati e 60 broker per cluster per cluster per cluster basati. KRaft

Fase 2: Configurare le impostazioni di archiviazione e cluster

Questa procedura descrive come configurare le esigenze di archiviazione dei dati tra tutti i broker e specificare la modalità di archiviazione. Questo ti aiuta a definire i requisiti di archiviazione dei dati in base alle esigenze del carico di lavoro. Inoltre, questa procedura descrive le impostazioni di configurazione del cluster che controllano il funzionamento dei broker. Queste impostazioni includono le configurazioni dei broker, le impostazioni degli argomenti predefinite e le politiche di archiviazione su più livelli.

1. Se hai selezionato il tipo di broker come Standard, procedi come segue nella sezione Archiviazione:

   a. Per Storage, scegli la quantità iniziale di spazio di archiviazione che desideri assegnare al cluster. Non è possibile ridurre la capacità di archiviazione dopo aver creato il cluster.

   b. (Facoltativo) A seconda della dimensione del broker (dimensione dell'istanza) selezionata, puoi anche specificare il throughput di storage fornito per broker. Questa opzione consente di allocare prestazioni di input e output (I/O) dedicate per i volumi Amazon EBS di ciascun broker.

      Per abilitare questa opzione, scegli la dimensione del broker (dimensione dell'istanza) kafka.m5.4xlarge o superiore per x86 e kafka.m7g.2xlarge o superiore per le istanze basate su Graviton. Quindi, seleziona la casella di controllo Enable provisioned storage throughput. Selezionando questa casella di controllo, è possibile impostare manualmente un minimo di 250 MiB al secondo di throughput. Ciò è utile per carichi di lavoro o applicazioni a uso intensivo di I/O che richiedono prestazioni di storage prevedibili e ad alta velocità. Per ulteriori informazioni, consulta ???.

c. Per la modalità di archiviazione Cluster, specifica in che modo i dati vengono archiviati e gestiti all'interno del cluster. Questa opzione determina il tipo e la configurazione dello storage utilizzato dai broker. Selezionare una delle seguenti opzioni:

   - Solo storage EBS: archivia tutti i dati relativi agli argomenti localmente sui volumi Amazon Elastic Block Store (Amazon EBS) collegati a ciascun broker. Scegli questa modalità per esigenze prestazionali coerenti e accesso rapido ai messaggi recenti.
   - Storage su più livelli e storage EBS: combina i dati locali di Amazon EBS con lo storage remoto ed economico per set di dati di grandi dimensioni in Amazon S3. Questa modalità riduce i costi di storage di Amazon EBS, supporta una conservazione dei dati più lunga e ridimensiona automaticamente lo storage senza intervento manuale. Scegli questa modalità quando desideri conservare i dati per periodi più lunghi a costi inferiori o prevedi che le tue esigenze di storage aumentino in modo significativo.

> **Note**
>
> Non è necessario gestire lo storage per i broker Express.

2. Per la configurazione del cluster, specifica una delle seguenti opzioni per definire il comportamento del cluster:

   - Configurazione predefinita di Amazon MSK: contiene un set predefinito di configurazioni ottimizzate per casi d'uso generici. Scegli questa opzione per una configurazione e una distribuzione rapide del cluster. Per informazioni sulle configurazioni di Amazon MSK, consulta la sezione Configurazione Amazon MSK Provisioned.
   - Configurazione personalizzata: consente di specificare le impostazioni del broker e dell'argomento. È possibile scegliere una configurazione personalizzata esistente dall'elenco o creare una nuova configurazione personalizzata. Scegli questa opzione per un controllo preciso per i tuoi broker, ad esempio l'ottimizzazione specifica delle prestazioni, le impostazioni di sicurezza e altro ancora.

3. Scegli Successivo per continuare.

Fase 3: Configurare le impostazioni di rete

La configurazione di rete definisce il modo in cui il cluster viene distribuito all'interno AWS dell'infrastruttura. Ciò include VPC, zone di disponibilità e sottoreti e gruppi di sicurezza che controllano la rete, la disponibilità e l'accesso.

1. Per le reti, procedi come segue:

   a. Scegli il VPC che desideri utilizzare per il cluster.

   b. In base al numero di zone di disponibilità selezionate in precedenza, specifica le zone di disponibilità e le sottoreti in cui verranno implementate i broker.

      Per i broker standard nella regione Stati Uniti occidentali (California settentrionale), sono necessarie due sottoreti in due zone di disponibilità diverse. Per altre regioni in cui è disponibile Amazon MSK, è possibile specificare due o tre sottoreti. Le sottoreti devono trovarsi tutte in zone di disponibilità differenti.

      Per i broker Express, sono necessarie tre sottoreti in tre diverse zone di disponibilità.

      Quando si crea un cluster MSK Provisioned, MSK distribuisce i nodi broker in modo uniforme sulle sottoreti specificate.

   c. Per i gruppi di sicurezza in Amazon EC2, scegli o crea uno o più gruppi di sicurezza a cui desideri consentire l'accesso al tuo cluster. Questi gruppi EC2 di sicurezza Amazon controllano il traffico in entrata e in uscita verso i tuoi broker. Ad esempio, i gruppi di sicurezza delle macchine client.

      Se specificate gruppi di sicurezza condivisi con voi, dovete assicurarvi di disporre delle autorizzazioni per utilizzarli. Nello specifico, è necessaria l'autorizzazione `ec2:DescribeSecurityGroups`. Per ulteriori informazioni, vedere Connessione a un cluster MSK.

2. Scegli Successivo per continuare.

Fase 4: Configurare le impostazioni di sicurezza

1. Nella sezione Impostazioni di sicurezza, procedi come segue:

   - Scegli uno o più dei seguenti metodi di autenticazione e autorizzazione per controllare l'accesso dei client ai tuoi cluster Kafka:

- Accesso non autenticato: consente ai client di accedere al cluster senza fornire alcuna credenziale di autenticazione. Questo metodo rappresenta un rischio per la sicurezza e potrebbe non essere conforme alle migliori pratiche di sicurezza. Per ulteriori informazioni, consulta msk-unrestricted-access-check.
- Autenticazione basata sui ruoli IAM: consente l'autenticazione e l'autorizzazione del client utilizzando utenti/ruoli AWS IAM. Questo metodo fornisce un controllo granulare sull'accesso ai cluster tramite le policy IAM. Abbiamo consigliato questo metodo per le applicazioni già in esecuzione in. AWS
- Autenticazione SASL/SCRAM: richiede ai client di fornire le credenziali di nome utente e password memorizzate per l'autenticazione. AWS Secrets Manager Amazon MSK recupera queste credenziali da Secrets Manager e autentica gli utenti in modo sicuro.

  Per configurare le credenziali di accesso relative all'autenticazione per un cluster, crea prima una risorsa segreta in Secrets Manager. Quindi, associa le credenziali di accesso a quel segreto. Per ulteriori informazioni su questo metodo di controllo dell'accesso, vedere. Configurare SASL/SCRAM l'autenticazione per un cluster Amazon MSK
- Autenticazione client TLS tramite AWS Certificate Manager (ACM): consente l'autenticazione reciproca tra client e broker utilizzando certificati digitali. È necessario configurare un AWS Private Certificate Authority (AWS Private CA) nello stesso o in un altro cluster Account AWS .

  Consigliamo vivamente di utilizzare AWS Private CA s indipendenti per ogni cluster MSK durante l'implementazione di MTL. Ciò garantisce che i certificati TLS firmati da si autentichino PCAs solo con un singolo cluster MSK, mantenendo così un rigoroso controllo degli accessi.

2. In Encryption, scegli il tipo di chiave KMS che desideri utilizzare per crittografare i dati inattivi. Per ulteriori informazioni, consulta the section called “Crittografia Amazon MSK a riposo”.

   La crittografia dei dati inattivi protegge l'integrità dei dati archiviati, mentre la crittografia in transito protegge la riservatezza dei dati dal monitoraggio della rete durante il trasferimento.
3. Scegli Successivo per continuare.

Fase 5: Configurare le opzioni di monitoraggio

Questa procedura descrive come impostare le metriche del broker e raccogliere e fornire i log del broker. Con queste impostazioni, puoi osservare e analizzare lo stato e le prestazioni del cluster e risolvere i problemi. Per ulteriori informazioni, consulta the section called “Monitora un cluster”.

1. Per i CloudWatch parametri Amazon per questo cluster, scegli uno dei seguenti livelli di monitoraggio. Le metriche raccolte a ciascun livello di monitoraggio sono integrate CloudWatch per la visualizzazione e gli avvisi.

   a. Monitoraggio di base: fornisce una serie di metriche essenziali a livello di cluster senza costi aggiuntivi. Questo livello è utile per la maggior parte dei casi d'uso con esigenze di monitoraggio generali.

   b. Monitoraggio avanzato a livello di broker: fornisce metriche dettagliate dei broker a costi aggiuntivi. Questo livello include il monitoraggio di base e parametri più granulari dei broker, come i parametri di storage su più livelli, i byte in/out di altri broker e il tempo totale di utilizzo delle operazioni. read/write Paghi per le metriche di questo livello, mentre le metriche di livello base continuano a essere gratuite.

   c. Monitoraggio avanzato a livello di argomento: fornisce metriche per singoli argomenti a un costo aggiuntivo. Scegli questo livello per ottenere una visione più dettagliata delle prestazioni degli argomenti tra i broker. Questo livello include il monitoraggio avanzato a livello di broker e parametri a livello di argomento, come i parametri di storage su più livelli per un argomento specifico e il numero di messaggi ricevuti al secondo.

   d. Monitoraggio avanzato a livello di partizione: fornisce la visualizzazione più granulare delle metriche per partizione a costi aggiuntivi. Scegli questo livello per ottenere il monitoraggio più dettagliato acquisendo le metriche per ogni partizione all'interno di ogni argomento tra i broker. Questo livello include un monitoraggio avanzato a livello di argomento e metriche dettagliate specifiche delle partizioni, come le metriche di offset lag.

   Per ulteriori informazioni sulle metriche disponibili per i tipi di broker Standard ed Express a ciascuno di questi livelli di monitoraggio, consulta e. CloudWatch metriche per i broker Standard CloudWatch metriche per i broker Express

2. (Facoltativo) Se desideri esportare le metriche in formato Prometheus utilizzando JMX Exporter, Node Exporter o entrambi, scegli Abilita il monitoraggio aperto con Prometheus. Per ulteriori informazioni su questa opzione, consulta Monitora con Prometheus.

3. (Facoltativo) Per configurare il cluster MSK in modo da fornire i log dei broker a vari utenti per la risoluzione dei problemi e il controllo, scegliete una o più delle Servizi AWS seguenti opzioni. Amazon MSK non crea queste risorse di destinazione se non esistono già. Per ulteriori informazioni, consulta Log di broker.

   - Consegna ad Amazon CloudWatch Logs: invia i log a CloudWatch con funzionalità di clustering, ricerca e visualizzazione. Puoi interrogare e analizzare i log senza uscire da. AWS Management Console
   - Consegna ad Amazon S3: archivia i log come file in bucket Amazon S3 per l'archiviazione a lungo termine e l'analisi in batch.
   - Consegna ad Amazon Data Firehose: invia i log a Firehose per la consegna automatica ad Amazon OpenSearch Service per la risoluzione dei problemi in tempo reale.

4. (Facoltativo) Per facilitare l'identificazione, l'organizzazione o la ricerca del cluster, scegli Aggiungi nuovo tag per aggiungere tag come coppie chiave-valore. Ad esempio, aggiungi un tag al cluster con la coppia chiave-valore e. **`Load testing`** **`Test`**

   Per ulteriori informazioni sull'utilizzo dei tag nei cluster, consulta. Contrassegna un tag a un cluster Amazon MSK

5. Scegli Successivo per continuare.

Fase 6: Rivedere la configurazione del cluster

1. Rivedi le impostazioni del tuo cluster.

   Scegli Modifica o Precedente per modificare le impostazioni specificate in precedenza o tornare alla schermata precedente della console.

2. Scegli Create cluster (Crea cluster).
3. Controlla lo stato di questo cluster nella sezione di riepilogo del cluster della pagina dei dettagli del cluster. Quando Amazon MSK assegna il cluster, lo stato passa da Creazione in corso ad Attivo. Quando lo stato è Attivo, puoi connetterti al cluster. Per ulteriori informazioni sugli stati del cluster, consulta la pagina Comprendi gli stati del cluster MSK Provisioned.

Crea un cluster Amazon MSK con provisioning utilizzando AWS CLI

1. Copiare il JSON seguente e salvarlo in un file. Assegnare un nome al file `brokernodegroupinfo.json`. Sostituisci la sottorete IDs in JSON con i valori che corrispondono alle sottoreti. Le sottoreti devono trovarsi in zone di disponibilità differenti. Sostituisci *"Security-Group-ID"* con l'ID di uno o più gruppi di sicurezza del VPC client. I client associati a questi gruppi di sicurezza ottengono l'accesso al cluster. Se specifichi gruppi di sicurezza condivisi con te, devi verificare di disporre delle autorizzazioni per gli stessi. Nello specifico, è necessaria l'autorizzazione `ec2:DescribeSecurityGroups`. Per un esempio, consulta Amazon EC2: consente la gestione dei gruppi EC2 di sicurezza Amazon associati a un VPC specifico, a livello di programmazione e nella console. Infine, salva il file JSON aggiornato sul computer su cui lo hai installato. AWS CLI

```
{
  "InstanceType": "kafka.m5.large",
  "ClientSubnets": [
    "Subnet-1-ID",
    "Subnet-2-ID"
  ],
  "SecurityGroups": [
    "Security-Group-ID"
  ]
}
```

> **Important**
>
> Per i broker Express, sono necessarie tre sottoreti in tre diverse zone di disponibilità. Inoltre, non è necessario definire alcuna proprietà relativa allo storage.
> Per i broker standard nella regione Stati Uniti occidentali (California settentrionale), sono necessarie due sottoreti in due diverse zone di disponibilità. Per altre regioni in cui è disponibile Amazon MSK, è possibile specificare due o tre sottoreti. Le sottoreti devono trovarsi tutte in zone di disponibilità differenti. Quando crei un cluster, Amazon MSK distribuisce i nodi dei broker in modo uniforme nelle sottoreti specificate.

2. Eseguite il AWS CLI comando seguente nella directory in cui avete salvato il `brokernodegroupinfo.json` file, sostituendolo *"Your-Cluster-Name"* con un nome a vostra scelta. Infatti*"Monitoring-Level"*, è possibile specificare uno dei tre valori seguenti:`DEFAULT`,`PER_BROKER`, o`PER_TOPIC_PER_BROKER`. Per informazioni su questi tre

diversi livelli di monitoraggio, consulta ???. Il parametro `enhanced-monitoring` è facoltativo. Se non viene specificato nel comando `create-cluster`, si ottiene il livello di monitoraggio `DEFAULT`.

```
aws kafka create-cluster --cluster-name "Your-Cluster-Name" --broker-node-group-
info file://brokernodegroupinfo.json --kafka-version "2.8.1" --number-of-broker-
nodes 3 --enhanced-monitoring "Monitoring-Level"
```

L'output del comando è simile al JSON seguente:

```
{
    "ClusterArn": "...",
    "ClusterName": "AWSKafkaTutorialCluster",
    "State": "CREATING"
}
```

> **Note**
>
> Il comando `create-cluster` potrebbe restituire un errore che indica che una o più sottoreti appartengono a zone di disponibilità non supportate. Quando ciò si verifica, l'errore indica quali zone di disponibilità non sono supportate. Crea sottoreti che non utilizzano le zone di disponibilità non supportate e riprova a eseguire nuovamente il comando `create-cluster`.

3. Salvare il valore della chiave `ClusterArn` perché è necessario per eseguire altre operazioni nel cluster.
4. Eseguire il comando seguente per verificare il tuo cluster `STATE`. Il valore `STATE` cambia da `CREATING` a `ACTIVE` quando Amazon EMR assegna il cluster. Quando lo stato è `ACTIVE`, puoi connetterti al cluster. Per ulteriori informazioni sugli stati del cluster, consulta la pagina Comprendi gli stati del cluster MSK Provisioned.

```
aws kafka describe-cluster --cluster-arn <your-cluster-ARN>
```

Crea un cluster MSK Provisioned con una configurazione Amazon MSK personalizzata utilizzando AWS CLI

Per informazioni sulle configurazioni personalizzate di Amazon MSK e su come crearle, consulta la sezione the section called “Configurazione del broker”.

1. Salva il seguente codice JSON in un file, sostituendolo *configuration-arn* con l'ARN della configurazione che desideri utilizzare per creare il cluster.

```
{
    "Arn": configuration-arn,
    "Revision": 1
}
```

2. Esegui il comando `create-cluster` e utilizza l'opzione `configuration-info` per puntare al file JSON salvato nella fase precedente. Di seguito è riportato un esempio.

```
aws kafka create-cluster --cluster-name ExampleClusterName --broker-node-group-
info file://brokernodegroupinfo.json --kafka-version "2.8.1" --number-of-broker-
nodes 3 --enhanced-monitoring PER_TOPIC_PER_BROKER --configuration-info file://
configuration.json
```

Di seguito è riportato un esempio di una risposta corretta dopo l'esecuzione di questo comando.

```
{
    "ClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/
CustomConfigExampleCluster/abcd1234-abcd-dcba-4321-a1b2abcd9f9f-2",
    "ClusterName": "CustomConfigExampleCluster",
    "State": "CREATING"
}
```

Crea un cluster MSK Provisioned utilizzando l'API Amazon MSK

L'API Amazon MSK ti consente di creare e gestire in modo programmatico il tuo cluster MSK Provisioned come parte di script di provisioning o distribuzione automatizzati dell'infrastruttura.

Per creare un cluster MSK Provisioned utilizzando l'API, consulta. CreateCluster

Elenca i cluster Amazon MSK

Per ottenere un broker di bootstrap per un cluster Amazon MSK, è necessario il cluster Amazon Resource Name (ARN). Se non disponi dell'ARN per il cluster, puoi trovarlo elencando tutti i cluster. Consultare the section called “Ottieni i broker bootstrap”.

Argomenti

- Elenca i cluster utilizzando il AWS Management Console
- Elenca i cluster utilizzando il AWS CLI
- Elenca i cluster utilizzando l'API

Elenca i cluster utilizzando il AWS Management Console

Per ottenere un broker di bootstrap per un cluster Amazon MSK, è necessario il cluster Amazon Resource Name (ARN). Se non disponi dell'ARN per il cluster, puoi trovarlo elencando tutti i cluster. Consultare the section called “Ottieni i broker bootstrap”.

1. Accedere a e aprire la console Amazon MSK da https://console.aws.amazon.com/msk/casa? AWS Management Console region=us-east-1#/home/.
2. La tabella mostra tutti i cluster per la regione corrente in questo account. Scegli il nome di un cluster per visualizzarne i dettagli.

Elenca i cluster utilizzando il AWS CLI

Per ottenere un broker di bootstrap per un cluster Amazon MSK, è necessario il cluster Amazon Resource Name (ARN). Se non disponi dell'ARN per il cluster, puoi trovarlo elencando tutti i cluster. Consultare the section called “Ottieni i broker bootstrap”.

```
aws kafka list-clusters
```

Elenca i cluster utilizzando l'API

Per ottenere un broker di bootstrap per un cluster Amazon MSK, è necessario il cluster Amazon Resource Name (ARN). Se non disponi dell'ARN per il cluster, puoi trovarlo elencando tutti i cluster. Consultare the section called “Ottieni i broker bootstrap”.

Per elencare i cluster che utilizzano l'API, consulta. ListClusters

Connect a un cluster Amazon MSK Provisioned

Per impostazione predefinita, i client possono accedere a un cluster MSK Provisioned solo se si trovano nello stesso VPC del cluster. Tutte le comunicazioni tra i client Kafka e il cluster MSK Provisioned sono private per impostazione predefinita e i dati di streaming non attraversano mai Internet. Per connetterti al tuo cluster MSK Provisioned da un client che si trova nello stesso VPC del cluster, assicurati che il gruppo di sicurezza del cluster disponga di una regola in entrata che accetti il traffico dal gruppo di sicurezza del client. Per informazioni sull'impostazione di queste regole, consulta Regole del gruppo di sicurezza. Per un esempio di come accedere a un cluster da un' EC2istanza Amazon che si trova nello stesso VPC del cluster, vedi. the section called “Inizia a usare”

> **Note**
>
> KRaft la modalità metadati e i broker MSK Express non possono abilitare sia il monitoraggio aperto che l'accesso pubblico.

Per connettersi al cluster MSK Provisioned da un client esterno al VPC del cluster, vedere Accesso dall'interno AWS ma dall'esterno del VPC del cluster.

Argomenti

- Attiva l'accesso pubblico a un cluster MSK Provisioned
- Accesso dall'interno AWS ma dall'esterno del VPC del cluster

Attiva l'accesso pubblico a un cluster MSK Provisioned

Amazon MSK ti offre la possibilità di attivare l'accesso pubblico ai broker dei cluster MSK Provisioned che eseguono Apache Kafka 2.6.0 o versioni successive. Per motivi di sicurezza, non è possibile attivare l'accesso pubblico durante la creazione di un cluster MSK. Tuttavia, è possibile aggiornare un cluster esistente per renderlo accessibile al pubblico. È inoltre possibile creare un nuovo cluster e aggiornarlo in modo da renderlo accessibile al pubblico.

Puoi attivare l'accesso pubblico a un cluster MSK senza costi aggiuntivi, ma per il trasferimento dei dati da e verso il cluster si applicano i costi standard di trasferimento AWS dei dati. Per informazioni sui prezzi, consulta la pagina dei prezzi di Amazon EC2 On-Demand.

> **Note**
>
> Se utilizzi i metodi di controllo degli accessi SASL/SCRAM o mTLS, devi prima impostare Apache Kafka per il tuo cluster. ACLs Quindi, aggiorna la configurazione del cluster per impostare la proprietà su false. `allow.everyone.if.no.acl.found` Per informazioni su come aggiornare la configurazione di un cluster, consulta la pagina the section called "Operazioni di configurazione del broker".

Per attivare l'accesso pubblico a un cluster MSK Provisioned, assicuratevi che il cluster soddisfi tutte le seguenti condizioni:

- Le sottoreti associate al cluster devono essere pubbliche. A ogni sottorete pubblica è associato un IPv4 indirizzo pubblico e il prezzo IPv4 degli indirizzi pubblici è indicato nella pagina dei prezzi di Amazon VPC. Ciò significa che le sottoreti devono avere una tabella di routing associata a un gateway Internet. Per informazioni su come creare e collegare un gateway Internet, consulta Abilita l'accesso a Internet VPC utilizzando i gateway Internet nella Amazon VPC User Guide.
- Il controllo degli accessi non autenticati deve essere disattivato e almeno uno dei seguenti metodi di controllo degli accessi deve essere attivo:, MTL. SASL/IAM, SASL/SCRAM Per informazioni su come aggiornare il metodo di controllo degli accessi di un cluster, consulta la pagina the section called "Aggiorna la sicurezza del cluster".
- La crittografia all'interno del cluster deve essere attiva. Per impostazione predefinita durante la creazione di un cluster, la crittografia è attiva. Non è possibile attivare la crittografia all'interno del cluster se esso è stato creato con questa opzione disattivata. Pertanto, non è possibile attivare l'accesso pubblico per il cluster se esso è stato creato con la crittografia all'interno del cluster disattivata.
- Il traffico non crittografato tra broker e client deve essere disattivato. Per informazioni su come disattivarlo se è attivato, consulta la pagina the section called "Aggiorna la sicurezza del cluster".
- Se utilizzi il controllo degli accessi IAM e desideri applicare politiche di autorizzazione o aggiornare le politiche di autorizzazione, consulta. the section called "Controllo degli accessi IAM" Per informazioni su Apache Kafka ACLs, consulta. the section called "Apache Kafka ACLs"

Dopo esserti assicurato che un cluster MSK soddisfi le condizioni sopra elencate AWS Management Console, puoi utilizzare l' AWS CLI API Amazon MSK per attivare l'accesso pubblico. Dopo aver attivato l'accesso pubblico a un cluster, puoi recuperare una stringa bootstrap-brokers pubblica

relativa al cluster. Per informazioni su come recuperare i broker di bootstrap per un cluster, consulta la pagina the section called “Ottieni i broker bootstrap”.

> **Important**
>
> Oltre ad attivare l'accesso pubblico, assicurati che i gruppi di sicurezza del cluster dispongano di regole TCP in entrata che consentano l'accesso pubblico dal tuo indirizzo IP. Ti consigliamo di impostare tali regole di modo che siano il più restrittive possibile. Per ulteriori informazioni sui gruppi di sicurezza e le regole in entrata, consulta la pagina Security groups for your VPC nella Guida per l'utente di Amazon VPC. Per i numeri di porta, consulta la pagina the section called “Informazioni sulle porte”. Per istruzioni su come modificare il gruppo di sicurezza di un cluster, consulta la pagina the section called “Modifica dei gruppi di sicurezza”.

> **Note**
>
> Se dopo avere seguito le istruzioni seguenti per attivare l'accesso pubblico non riesci comunque ad accedere al cluster, consulta la pagina the section called “Impossibile accedere al cluster con accesso pubblico attivato”.

Attivazione dell'accesso pubblico tramite la console

1. Accedere a e aprire la console Amazon MSK da https://console.aws.amazon.com/msk/casa?AWS Management Console region=us-east-1#/home/.
2. Nell'elenco dei cluster, scegli quello per il quale attivare l'accesso pubblico.
3. Scegli la scheda Proprietà, quindi trova la sezione Impostazioni di rete.
4. Scegli Modifica accesso pubblico.

Attivazione dell'accesso pubblico tramite AWS CLI

1. Esegui il AWS CLI comando seguente, sostituendo *`ClusterArn`* e *`Current-Cluster-Version`* con l'ARN e la versione corrente del cluster. Per trovare la versione corrente del cluster, usa l'DescribeClusteroperazione o il comando AWS CLI describe-cluster. Una versione di esempio è `KTVPDKIKX0DER`.

```
aws kafka update-connectivity --cluster-arn ClusterArn --current-
version Current-Cluster-Version --connectivity-info '{"PublicAccess": {"Type":
 "SERVICE_PROVIDED_EIPS"}}'
```

L'output di questo comando `update-connectivity` è simile all'esempio JSON seguente.

```
{
    "ClusterArn": "arn:aws:kafka:us-east-1:012345678012:cluster/exampleClusterName/
abcdefab-1234-abcd-5678-cdef0123ab01-2",
    "ClusterOperationArn": "arn:aws:kafka:us-east-1:012345678012:cluster-
operation/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2/0123abcd-
abcd-4f7f-1234-9876543210ef"
}
```

> **Note**
>
> Per disattivare l'accesso pubblico, usa un AWS CLI comando simile, ma con le seguenti informazioni di connettività:
>
> ```
> '{"PublicAccess": {"Type": "DISABLED"}}'
> ```

2. Per ottenere il risultato dell'`update-connectivity`operazione, esegui il comando seguente, sostituendolo *`ClusterOperationArn`* con l'ARN ottenuto nell'output del `update-connectivity` comando.

```
aws kafka describe-cluster-operation --cluster-operation-arn ClusterOperationArn
```

L'output di questo comando `describe-cluster-operation` è simile all'esempio JSON seguente.

```
{
    "ClusterOperationInfo": {
        "ClientRequestId": "982168a3-939f-11e9-8a62-538df00285db",
        "ClusterArn": "arn:aws:kafka:us-east-1:012345678012:cluster/
exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2",
        "CreationTime": "2019-06-20T21:08:57.735Z",
```

```
        "OperationArn": "arn:aws:kafka:us-east-1:012345678012:cluster-
operation/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2/0123abcd-
abcd-4f7f-1234-9876543210ef",
        "OperationState": "UPDATE_COMPLETE",
        "OperationType": "UPDATE_CONNECTIVITY",
        "SourceClusterInfo": {
            "ConnectivityInfo": {
                "PublicAccess": {
                    "Type": "DISABLED"
                }
            }
        },
        "TargetClusterInfo": {
            "ConnectivityInfo": {
                "PublicAccess": {
                    "Type": "SERVICE_PROVIDED_EIPS"
                }
            }
        }
    }
}
```

Se il valore di `OperationState` è `UPDATE_IN_PROGRESS`, attendi qualche minuto, quindi esegui nuovamente il comando `describe-cluster-operation`.

Attivazione dell'accesso pubblico tramite l'API di Amazon MSK

- Per utilizzare l'API per attivare o disattivare l'accesso pubblico a un cluster, consulta UpdateConnectivity.

> **Note**
>
> Per motivi di sicurezza, Amazon MSK non consente l'accesso pubblico ad Apache ZooKeeper o ai nodi KRaft del controller.

Accesso dall'interno AWS ma dall'esterno del VPC del cluster

Per connettersi a un cluster MSK dall'interno AWS ma dall'esterno dell'Amazon VPC del cluster, esistono le seguenti opzioni.

Peering Amazon VPC

Per connetterti al tuo cluster MSK da un VPC diverso dal VPC del cluster, puoi creare una connessione peering tra i due. VPCs Per informazioni sul peering VPC, consulta la Guida al peering di VPC di Amazon.

AWS Direct Connect

AWS Direct Connect collega la rete locale a un cavo in fibra ottica AWS Ethernet standard da 1 o 10 gigabit. Un'estremità del cavo è collegata al router, l'altra a un router. AWS Direct Connect Con questa connessione, puoi creare interfacce virtuali direttamente sul AWS cloud e Amazon VPC, aggirando i provider di servizi Internet nel tuo percorso di rete. Per ulteriori informazioni, consulta AWS Direct Connect.

AWS Transit Gateway

AWS Transit Gateway è un servizio che ti consente di connettere le tue reti VPCs e quelle locali a un unico gateway. Per informazioni su come utilizzare AWS Transit Gateway, consulta AWS Transit Gateway.

Connessioni VPN

Puoi connettere il VPC del cluster MSK a reti e utenti remoti utilizzando le opzioni di connettività VPN descritte nel seguente argomento: VPN Connections.

Proxy REST

Puoi installare un proxy REST in un'istanza in esecuzione all'interno dell'Amazon VPC del cluster. I proxy REST consentono ai produttori e ai consumatori di comunicare con il cluster attraverso richieste API HTTP.

Connettività multi-VPC per regioni multiple

Il documento seguente descrive le opzioni di connettività per più regioni VPCs che risiedono in regioni diverse: Connettività multi-VPC a più regioni.

Connettività privata multi-VPC a regione singola

La connettività privata multi-VPC (con tecnologia AWS PrivateLink) per i cluster Amazon Managed Streaming for Apache Kafka (Amazon MSK) è una funzionalità che consente di connettere più

rapidamente i client Kafka ospitati in diversi Virtual VPCs Private Cloud () e account a un cluster Amazon MSK. AWS

Consulta la sezione Single Region multi-VPC connectivity for cross-account clients.

EC2-La rete classica è stata ritirata

Amazon MSK non supporta più EC2 le istanze Amazon in esecuzione con reti Amazon EC2 -Classic.

Vedi EC2-Classic Networking si sta ritirando: ecco come prepararsi.

Connettività privata multi-VPC di Amazon MSK in un'unica regione

La connettività privata multi-VPC (con tecnologia AWS PrivateLink) per i cluster Amazon Managed Streaming for Apache Kafka (Amazon MSK) è una funzionalità che consente di connettere più rapidamente i client Kafka ospitati in diversi Virtual VPCs Private Cloud () e account a un cluster Amazon MSK. AWS

La connettività privata multi-VPC è una soluzione gestita che semplifica l'infrastruttura di rete per la connettività multi-VPC e multi-account. I client possono connettersi al cluster Amazon MSK PrivateLink mantenendo tutto il traffico all'interno della AWS rete. La connettività privata multi-VPC per i cluster Amazon MSK è disponibile in tutte le regioni in AWS cui è disponibile Amazon MSK.

Argomenti

Cos'è la connettività privata multi-VPC?

La connettività privata multi-VPC per Amazon MSK è un'opzione di connettività che consente di connettere client Apache Kafka ospitati in diversi account e AWS cloud privati virtuali (VPCs) a un cluster MSK.

Amazon MSK semplifica l'accesso multi-account con le policy del cluster. Queste politiche consentono al proprietario del cluster di concedere autorizzazioni ad altri AWS account per stabilire una connettività privata al cluster MSK.

Vantaggi della connettività privata multi-VPC

La connettività privata multi-VPC presenta diversi vantaggi rispetto ad altre soluzioni di connettività:

- Automatizza la gestione operativa della soluzione di connettività. AWS PrivateLink
- Consente la sovrapposizione IPs tra le connessioni VPCs, eliminando la necessità di mantenere tabelle di peering e routing complesse e non sovrapposte IPs associate ad altre soluzioni di connettività VPC.

Si utilizza una politica di cluster per il cluster MSK per definire quali AWS account dispongono delle autorizzazioni per configurare la connettività privata tra account verso il cluster MSK. L'amministratore multi-account può delegare le autorizzazioni ai ruoli o agli utenti appropriati. In combinazione con l'autenticazione del client IAM, puoi utilizzare la policy del cluster anche per definire in modo granulare le autorizzazioni del piano dati Kafka per i client che si connettono.

Requisiti e limitazioni per la connettività privata multi-VPC

Tieni conto di questi requisiti del cluster MSK per l'esecuzione della connettività privata multi-VPC:

- La connettività privata multi-VPC è supportata solo su Apache Kafka 2.7.1 o versioni successive. Assicurati che tutti i client utilizzati con il cluster MSK eseguano versioni di Apache Kafka compatibili con il cluster.
- La connettività privata multi-VPC supporta i tipi di autenticazione IAM, TLS e SASL/SCRAM. I cluster non autenticati non possono utilizzare la connettività privata multi-VPC.
- Se si utilizzano i metodi di controllo degli accessi SASL/SCRAM o MTLS, è necessario impostare Apache Kafka per il cluster. ACLs Innanzitutto, imposta Apache Kafka per il tuo cluster. ACLs Quindi, aggiorna la configurazione del cluster in modo che la proprietà `allow.everyone.if.no.acl.found` sia impostata su false per il cluster. Per informazioni su come aggiornare la configurazione di un cluster, consulta la pagina the section called “Operazioni di configurazione del broker”. Se utilizzi il Controllo degli accessi IAM e desideri applicare policy di autorizzazione o aggiornare le tue policy esistenti, consulta la sezione the section called “Controllo degli accessi IAM”. Per informazioni su Apache Kafka, consulta. ACLs the section called “Apache Kafka ACLs”
- La connettività privata multi-VPC non supporta il tipo di istanza t3.small.

- La connettività privata multi-VPC non è supportata in tutte AWS le regioni, ma solo negli AWS account all'interno della stessa regione.
- Per configurare la connettività privata multi-VPC, è necessario disporre dello stesso numero di sottoreti client delle sottoreti del cluster. È inoltre necessario assicurarsi che la zona IDs di disponibilità sia la stessa per la sottorete client e la sottorete del cluster.
- Amazon MSK non supporta la connettività privata multi-VPC ai nodi ZooKeeper.

Inizia a usare la connettività privata multi-VPC

Argomenti

- Passaggio 1: sul cluster MSK nell'account A, attiva la connettività multi-VPC per lo schema di autenticazione IAM sul cluster
- Passaggio 2: collegamento di una policy del cluster al cluster MSK
- Passaggio 3: operazioni dell'utente multi-account per configurare connessioni VPC gestite dal client

Questo tutorial utilizza un caso d'uso comune come esempio di come utilizzare la connettività multi-VPC per connettere privatamente un client Apache Kafka a un cluster MSK dall'interno AWS ma dall'esterno del VPC del cluster. Questo processo richiede che l'utente multi-account crei una connessione e una configurazione VPC gestite da MSK per ogni client, comprese le autorizzazioni client richieste. Il processo richiede inoltre che il proprietario del cluster MSK abiliti la PrivateLink connettività sul cluster MSK e selezioni gli schemi di autenticazione per controllare l'accesso al cluster.

In diverse parti di questo tutorial, scegliamo le opzioni che si applicano a questo esempio. Ciò non significa che siano le uniche opzioni che funzionano per la configurazione di un cluster MSK o delle istanze client.

La configurazione di rete per questo caso d'uso è la seguente:

- Un utente multi-account (client Kafka) e un cluster MSK si trovano nella stessa rete/regione AWS , ma in account diversi:
  - Cluster MSK nell'account A
  - Cliente Kafka nell'account B
- L'utente multi-account si connetterà privatamente al cluster MSK utilizzando lo schema di autenticazione IAM.

Questo tutorial presuppone che esista un cluster MSK assegnato creato con Apache Kafka versione 2.7.1 o successiva. Il cluster MSK deve essere in uno stato ACTIVE prima di iniziare il processo di configurazione. Per evitare potenziali perdite di dati o tempi di inattività, i client che utilizzeranno una connessione privata multi-VPC per connettersi al cluster devono utilizzare versioni di Apache Kafka compatibili con il cluster.

Il diagramma seguente illustra l'architettura della connettività multi-VPC di Amazon MSK connessa a un client in un account diverso. AWS

Passaggio 1: sul cluster MSK nell'account A, attiva la connettività multi-VPC per lo schema di autenticazione IAM sul cluster

Il proprietario del cluster MSK deve configurare le impostazioni di configurazione sul cluster MSK dopo la creazione del cluster e in uno stato ACTIVE.

Il proprietario del cluster attiva la connettività privata multi-VPC sul cluster ACTIVE per tutti gli schemi di autenticazione che saranno attivi sul cluster. Questa operazione può essere eseguita utilizzando l'UpdateSecurity API o la console MSK. La connettività privata multi-VPC supporta gli schemi di autenticazione IAM, TLS e SASL/SCRAM. La connettività privata multi-VPC non può essere abilitata per i cluster non autenticati.

In questo caso d'uso, configurerai il cluster per utilizzare lo schema di autenticazione IAM.

> **Note**
>
> Se state configurando il cluster MSK per utilizzare lo schema di SASL/SCRAM autenticazione, la proprietà Apache Kafka ACLs "" è obbligatoria. `allow.everyone.if.no.acl.found=false` Vedi ACLs Apache Kafka.

Quando aggiorni le impostazioni di connettività privata multi-VPC, Amazon MSK intraprende un riavvio progressivo dei nodi del broker che aggiorna le configurazioni del broker. Il completamento del processo può richiedere fino a 30 minuti o più. Non è possibile apportare altri aggiornamenti al cluster durante l'aggiornamento della connettività.

Attivazione del multi-VPC per gli schemi di autenticazione selezionati sul cluster nell'account A tramite la console

1. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/per l'account in cui si trova il cluster.
2. Nel riquadro di navigazione, in Cluster MSK, scegli Cluster per visualizzare l'elenco dei cluster presenti nell'account.
3. Seleziona il cluster da configurare per la connettività privata multi-VPC. Il cluster deve essere in uno stato ACTIVE.
4. Seleziona la scheda Proprietà del cluster, quindi vai a Impostazioni di rete.
5. Seleziona il menu a discesa Modifica e seleziona Attiva la connettività multi-VPC.
6. Seleziona uno o più tipi di autenticazione che desideri attivare per questo cluster. Per questo caso d'uso, seleziona l'autenticazione basata sui ruoli IAM.
7. Seleziona Salva modifiche.

Example - UpdateConnectivity API che attiva schemi di autenticazione della connettività privata multi-VPC su un cluster

In alternativa alla console MSK, è possibile utilizzare l'UpdateConnectivity API per attivare la connettività privata multi-VPC e configurare gli schemi di autenticazione su un cluster ACTIVE. L'esempio seguente mostra lo schema di autenticazione IAM attivato per il cluster.

```
{
  "currentVersion": "K3T4TT2Z381HKD",
  "connectivityInfo": {
    "vpcConnectivity": {
      "clientAuthentication": {
        "sasl": {
          "iam": {
            "enabled": TRUE
            }
        }
      }
    }
  }
}
```

Amazon MSK crea l'infrastruttura di rete necessaria per la connettività privata. Amazon MSK crea anche un nuovo set di endpoint broker di bootstrap per ogni tipo di autenticazione che richiede la connettività privata. Tieni presente che lo schema di autenticazione non crittografata non supporta la connettività privata multi-VPC.

Passaggio 2: collegamento di una policy del cluster al cluster MSK

Il proprietario del cluster può collegare una policy del cluster (nota anche come policy basata sulle risorse) al cluster MSK in cui verrà attivata la connettività privata multi-VPC. La policy del cluster fornisce ai client l'autorizzazione ad accedere al cluster da un altro account. Prima di poter modificare la policy del cluster, sono necessari gli ID degli account che devono disporre dell'autorizzazione ad accedere al cluster MSK. Consulta la sezione Funzionamento di Amazon MSK con IAM.

Il proprietario del cluster deve collegare al cluster MSK una policy del cluster che autorizzi l'utente multi-account nell'account B a recuperare i broker di bootstrap per il cluster e ad autorizzare le seguenti operazioni sul cluster MSK nell'account A:

- CreateVpcConnection
- GetBootstrapBrokers
- DescribeCluster
- DescribeClusterV2

Example

A titolo di riferimento, di seguito è riportato un esempio di JSON per una policy di cluster di base, simile alla policy predefinita mostrata nell'editor di policy IAM della console MSK. La seguente politica concede le autorizzazioni per l'accesso a livello di cluster, argomento e gruppo.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "123456789012"
      },
```

```
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:GetBootstrapBrokers",
        "kafka:DescribeCluster",
        "kafka:DescribeClusterV2",
        "kafka-cluster:*"
      ],
      "Resource": "arn:aws:kafka:us-east-1:111122223333:cluster/testing/
de8982fa-8222-4e87-8b20-9bf3cdfa1521-2"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "123456789012"
      },
      "Action": "kafka-cluster:*",
      "Resource": "arn:aws:kafka:us-east-1:111122223333:topic/testing/*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "123456789012"
      },
      "Action": "kafka-cluster:*",
      "Resource": "arn:aws:kafka:us-east-1:111122223333:group/testing/*"
    }
  ]
}
```

Collegamento di una policy del cluster al cluster MSK

1. Nella console Amazon MSK, in Cluster MSK, scegli Cluster.
2. Scorri verso il basso fino a Impostazioni di sicurezza e seleziona Modifica policy del cluster.
3. Nella console, nella schermata Modifica policy del cluster, seleziona Policy di base per la connettività multi-VPC.
4. Nel campo ID account, inserisci l'ID account per ogni account che dovrebbe disporre dell'autorizzazione per accedere a questo cluster. Durante la digitazione, l'ID viene automaticamente copiato nella sintassi JSON della policy visualizzata. Nel nostro esempio di policy del cluster, l'ID account è `123456789012`.
5. Seleziona Salva modifiche.

Per informazioni sulla politica dei cluster APIs, consulta le politiche basate sulle risorse di Amazon MSK.

Passaggio 3: operazioni dell'utente multi-account per configurare connessioni VPC gestite dal client

Per configurare la connettività privata multi-VPC tra un client in un account diverso dal cluster MSK, l'utente multi-account crea una connessione VPC gestita per il client. È possibile connettere più client al cluster MSK ripetendo questa procedura. Ai fini di questo caso d'uso, configurerai un solo client.

I client possono utilizzare gli schemi di autenticazione supportati IAM, SASL/SCRAM o TLS. A ogni connessione VPC gestita può essere associato un solo schema di autenticazione. Lo schema di autenticazione del client deve essere configurato nel cluster MSK a cui il client si connetterà.

In questo caso d'uso, configura lo schema di autenticazione del client in modo che il client nell'account B utilizzi lo schema di autenticazione IAM.

Prerequisiti

Questo processo richiede i seguenti elementi:

- La policy del cluster creata in precedenza che concede al client dell'account B l'autorizzazione a eseguire operazioni sul cluster MSK nell'account A.
- Una politica di identità allegata al client nell'Account B che concede autorizzazioni e azioni`kafka:CreateVpcConnection`. `ec2:CreateTags` `ec2:CreateVPCEndpoint` `ec2:DescribeVpcAttribute`

Example

A titolo di riferimento, di seguito è riportato un esempio di JSON per una policy di identità del client di base.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "kafka:CreateVpcConnection",
        "ec2:CreateTags",
        "ec2:CreateVPCEndpoint",
        "ec2:DescribeVpcAttribute"
      ],
      "Resource": "*"
    }
  ]
}
```

Creazione di una connessione VPC gestita per un client nell'account B

1. Dall'amministratore del cluster, ottieni l'ARN del cluster MSK nell'account A al quale desideri che il client nell'account B si connetta. Prendi nota dell'ARN del cluster da utilizzare in seguito.
2. Nella console MSK per l'account client B, scegli Connessioni VPC gestite, quindi scegli Crea connessione.
3. Nel riquadro Impostazioni di connessione, incolla l'ARN del cluster nel campo di testo ARN del cluster, quindi scegli Verifica.
4. Seleziona Tipo di autenticazione per il client nell'account B. Per questo caso d'uso, scegli IAM quando crei la connessione VPC del client.
5. Scegli il VPC per il client.
6. Scegli almeno due zone di disponibilità e sottoreti associate. È possibile ottenere la zona di disponibilità IDs dai dettagli del cluster della Console di AWS gestione o utilizzando l'DescribeClusterAPI o il comando AWS CLI describe-cluster. La zona specificata per la sottorete client deve corrispondere a IDs quella della sottorete del cluster. Se mancano i valori per una sottorete, crea innanzitutto una sottorete con lo stesso ID di zona del cluster MSK.
7. Scegli un Gruppo di sicurezza per questa connessione VPC. È possibile accettare il gruppo di sicurezza predefinito. Per ulteriori informazioni sulla configurazione di un gruppo di sicurezza, consulta la pagina Control traffic to resources using security groups.
8. Seleziona Crea connessione.
9. Per ottenere l'elenco delle nuove stringhe del broker di bootstrap dalla console MSK dell'utente multi-account (Dettagli del cluster > Connessione VPC gestita), consulta le stringhe del broker di bootstrap mostrate in "Stringa di connessione al cluster". Dall'account B del cliente, è possibile visualizzare l'elenco dei broker bootstrap richiamando l'GetBootstrapBrokersAPI o visualizzando l'elenco dei broker bootstrap nei dettagli del cluster della console.
10. Aggiorna i gruppi di sicurezza associati alle connessioni VPC come segue:

a. Imposta le regole in entrata per il PrivateLink VPC per consentire tutto il traffico per l'intervallo IP dalla rete dell'Account B.

b. [Facoltativo] Imposta la connettività delle Regole in uscita al cluster MSK. Scegli il Gruppo di sicurezza nella console VPC, Modifica le regole in uscita e aggiungi una regola per il Traffico TCP personalizzato per gli intervalli di porte 14001-14100. Il Network Load Balancer multi-VPC è in ascolto sugli intervalli di porte 14001-14100. Consulta la pagina Network Load Balancer.

11. Configura il client nell'account B per utilizzare i nuovi broker di bootstrap per la connettività privata multi-VPC per connettersi al cluster MSK nell'account A. Consulta la sezione Produzione e utilizzo di dati.

Una volta completata l'autorizzazione, Amazon MSK crea una connessione VPC gestita per ogni VPC e schema di autenticazione specificati. Il gruppo di sicurezza scelto è associato a ciascuna connessione. Questa connessione VPC gestita è configurata da Amazon MSK per connettersi privatamente ai broker. Puoi utilizzare il nuovo set di broker di bootstrap per connetterti privatamente al cluster Amazon MSK.

Aggiornamento degli schemi di autorizzazione su un cluster

La connettività privata multi-VPC supporta diversi schemi di autorizzazione: connettività SASL/SCRAM, IAM, and TLS. The cluster owner can turn on/off privata per uno o più schemi di autenticazione. Il cluster deve essere in stato ACTIVE per eseguire questa operazione.

Attivazione di uno schema di autenticazione tramite la console Amazon MSK

1. Apri la console Amazon MSK all'indirizzo AWS Management Console per il cluster che desideri modificare.
2. Nel riquadro di navigazione, in Cluster MSK, scegli Cluster per visualizzare l'elenco dei cluster presenti nell'account.
3. Seleziona il cluster da modificare. Il cluster deve essere in uno stato ACTIVE.
4. Seleziona la scheda Proprietà del cluster, quindi vai a Impostazioni di rete.
5. Seleziona il menu a discesa Modifica e seleziona Attiva la connettività multi-VPC per attivare un nuovo schema di autenticazione.
6. Seleziona uno o più tipi di autenticazione che desideri attivare per questo cluster.
7. Seleziona Attiva la selezione.

Quando attivi un nuovo schema di autenticazione, dovresti anche creare nuove connessioni VPC gestite per il nuovo schema di autenticazione e aggiornare i client di modo che utilizzino i broker di bootstrap specifici per il nuovo schema di autenticazione.

Disattivazione di uno schema di autenticazione tramite la console Amazon MSK

> **Note**
>
> Quando si disattiva la connettività privata multi-VPC per gli schemi di autenticazione, tutte le infrastrutture relative alla connettività, incluse le connessioni VPC gestite, vengono eliminate.

Quando si disattiva la connettività privata multi-VPC per gli schemi di autenticazione, le connessioni VPC esistenti sul lato client diventano INACTIVE e l'infrastruttura PrivateLink sul lato cluster, incluse le connessioni VPC gestite, viene rimossa. L'utente multi-account può eliminare solo la connessione VPC inattiva. Se sul cluster viene riattivata la connettività privata, l'utente multi-account deve creare una nuova connessione al cluster.

1. Apri la console Amazon MSK all'indirizzo AWS Management Console.
2. Nel riquadro di navigazione, in Cluster MSK, scegli Cluster per visualizzare l'elenco dei cluster presenti nell'account.
3. Seleziona il cluster da modificare. Il cluster deve essere in uno stato ACTIVE.
4. Seleziona la scheda Proprietà del cluster, quindi vai a Impostazioni di rete.
5. Seleziona il menu a discesa Modifica e seleziona Disattiva la connettività multi-VPC per disattivare uno schema di autenticazione.
6. Seleziona uno o più tipi di autenticazione che desideri disattivare per questo cluster.
7. Seleziona Disattiva la selezione.

Example Per attivare on/off uno schema di autenticazione con l'API

In alternativa alla console MSK, è possibile utilizzare l'UpdateConnectivity API per attivare la connettività privata multi-VPC e configurare gli schemi di autenticazione su un cluster ACTIVE. L'esempio seguente mostra gli SASL/SCRAM schemi di autenticazione IAM attivati per il cluster.

Quando attivi un nuovo schema di autenticazione, dovresti anche creare nuove connessioni VPC gestite per il nuovo schema di autenticazione e aggiornare i client di modo che utilizzino i broker di bootstrap specifici per il nuovo schema di autenticazione.

Quando si disattiva la connettività privata multi-VPC per gli schemi di autenticazione, le connessioni VPC esistenti sul lato client diventano INACTIVE e l'infrastruttura PrivateLink sul lato cluster, incluse le connessioni VPC gestite, viene rimossa. L'utente multi-account può eliminare solo la connessione VPC inattiva. Se sul cluster viene riattivata la connettività privata, l'utente multi-account deve creare una nuova connessione al cluster.

```
Request:
{
  "currentVersion": "string",
  "connnectivityInfo": {
    "publicAccess": {
      "type": "string"
    },
    "vpcConnectivity": {
      "clientAuthentication": {
        "sasl": {
          "scram": {
            "enabled": TRUE
          },
          "iam": {
            "enabled": TRUE
          }
        },
        "tls": {
          "enabled": FALSE
        }
      }
    }
  }
}

Response:
{
  "clusterArn": "string",
  "clusterOperationArn": "string"
}
```

Rifiuto di una connessione VPC gestita a un cluster Amazon MSK

Dalla console Amazon MSK sull'account amministratore del cluster, puoi rifiutare una connessione VPC client. La connessione VPC del client deve essere nello stato AVAILABLE per essere rifiutata. Potresti voler rifiutare una connessione VPC gestita da un client che non è più autorizzato a

connettersi al tuo cluster. Per evitare che nuove connessioni VPC gestite si connettano a un client, rifiuta l'accesso al client nella policy del cluster. Una connessione rifiutata comporta comunque dei costi fino a quando non viene eliminata dal proprietario della connessione. Consulta la sezione Eliminazione di una connessione VPC gestita a un cluster Amazon MSK.

Rifiuto di una connessione VPC client tramite la console MSK

1. Apri la console Amazon MSK all'indirizzo AWS Management Console.
2. Nel riquadro di navigazione, seleziona Cluster e scorri fino all'elenco Impostazioni di rete > Connessioni VPC client.
3. Seleziona la connessione che desideri rifiutare e seleziona Rifiuta connessione VPC client.
4. Conferma il rifiuto della connessione VPC client selezionata.

Per rifiutare una connessione VPC gestita tramite l'API, utilizza l'API `RejectClientVpcConnection`.

Eliminazione di una connessione VPC gestita a un cluster Amazon MSK

L'utente multi-account può eliminare una connessione VPC gestita per un cluster MSK dalla console dell'account client. Poiché l'utente proprietario del cluster non possiede la connessione VPC gestita, la connessione non può essere eliminata dall'account amministratore del cluster. Una volta eliminata, una connessione VPC non comporta più costi.

Eliminazione di una connessione VPC tramite la console MSK

1. Dall'account client, apri la console Amazon MSK all'indirizzo AWS Management Console.
2. Nel riquadro di navigazione, seleziona Connessioni VPC gestite.
3. Dall'elenco delle connessioni, seleziona la connessione VPN da eliminare.
4. Conferma l'eliminazione della connessione VPC.

Per eliminare una connessione VPC gestita tramite l'API, utilizza l'API `DeleteVpcConnection`.

Autorizzazioni per la connettività privata multi-VPC

Questa sezione riassume le autorizzazioni necessarie per client e cluster che utilizzano la funzionalità di connettività privata multi-VPC. La connettività privata multi-VPC richiede che l'amministratore del client crei le autorizzazioni su ogni client che avrà una connessione VPC gestita al cluster MSK.

Richiede inoltre che l'amministratore del cluster MSK abiliti la PrivateLink connettività sul cluster MSK e selezioni gli schemi di autenticazione per controllare l'accesso al cluster.

Autenticazione del cluster e autorizzazioni di accesso all'argomento

Attiva la funzionalità di connettività privata multi-VPC per gli schemi di autenticazione abilitati per il tuo cluster MSK. Consultare Requisiti e limitazioni per la connettività privata multi-VPC. Se si sta configurando il cluster MSK per utilizzare lo schema di SASL/SCRAM autenticazione, la proprietà Apache Kafka è obbligatoria. ACLs `allow.everyone.if.no.acl.found=false` Dopo avere impostato le Apache Kafka ACLs per il cluster, aggiorna la configurazione del cluster in modo che la proprietà `allow.everyone.if.no.acl.found` del cluster sia impostata su false. Per informazioni su come aggiornare la configurazione di un cluster, consulta la pagina Operazioni di configurazione del broker.

Autorizzazioni delle policy del cluster multi-account

Se un client Kafka si trova in un AWS account diverso dal cluster MSK, allega al cluster MSK una policy basata su cluster che autorizzi l'utente root del client per la connettività tra account. È possibile modificare la policy del cluster multi-VPC utilizzando l'editor di policy IAM nella console MSK (impostazioni di sicurezza del cluster > Modifica policy del cluster) oppure utilizzare quanto segue APIs per gestire la policy del cluster:

PutClusterPolicy

Collega la policy del cluster al cluster. È possibile utilizzare questa API per creare o aggiornare la policy del cluster MSK specificata. Se stai aggiornando la policy, il campo CurrentVersion è obbligatorio nel payload della richiesta.

GetClusterPolicy

Recupera il testo JSON del documento di policy del cluster collegato al cluster.

DeleteClusterPolicy

Elimina la policy del cluster.

Di seguito è riportato un esempio di JSON per una policy di cluster di base, simile a quella mostrata nell'editor di policy IAM della console MSK. La seguente policy concede le autorizzazioni per l'accesso a livello di cluster, argomento e gruppo.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {
            "AWS": [
                "123456789012"
            ]
        },
        "Action": [
            "kafka-cluster:*",
            "kafka:CreateVpcConnection",
            "kafka:GetBootstrapBrokers",
            "kafka:DescribeCluster",
            "kafka:DescribeClusterV2"
        ],
        "Resource": [
            "arn:aws:kafka:us-east-1:123456789012:cluster/testing/
de8982fa-8222-4e87-8b20-9bf3cdfa1521-2",
            "arn:aws:kafka:us-east-1:123456789012:topic/testing/*",
            "arn:aws:kafka:us-east-1:123456789012:group/testing/*"
        ]
    }]
}
```

Autorizzazioni client per la connettività privata multi-VPC a un cluster MSK

Per configurare la connettività privata multi-VPC tra un client Kafka e un cluster MSK, il client richiede una policy di identità collegata che conceda autorizzazioni per le operazioni `kafka:CreateVpcConnection`, `ec2:CreateTags` e `ec2:CreateVPCEndpoint` sul client. A titolo di riferimento, di seguito è riportato un esempio di JSON per una policy di identità del client di base.

JSON

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

```
    {
      "Effect": "Allow",
      "Action": [
        "kafka:CreateVpcConnection",
        "ec2:CreateTags",
        "ec2:CreateVPCEndpoint"
      ],
      "Resource": "*"
    }
  ]
}
```

Informazioni sulle porte

Utilizza i seguenti numeri di porta in modo che Amazon MSK possa comunicare con i computer client:

- Per comunicare con i broker con testo non crittografato, utilizza la porta 9092.
- Per comunicare con i broker con crittografia TLS, utilizza la porta 9094 per l'accesso dall'interno AWS e la porta 9194 per l'accesso pubblico.
- Per comunicare con i broker con SASL/SCRAM, utilizzate la porta 9096 per l'accesso dall'interno e la porta 9196 per l'accesso pubblico. AWS
- Per comunicare con i broker in un cluster configurato per l'usothe section called “Controllo degli accessi IAM”, utilizzate la porta 9098 per l'accesso dall'interno e la porta 9198 per l'accesso pubblico. AWS
- Per comunicare con Apache ZooKeeper utilizzando la crittografia TLS, utilizzate la porta 2182. ZooKeeper I nodi Apache utilizzano la porta 2181 per impostazione predefinita.

Ottieni i broker bootstrap per un cluster Amazon MSK

I broker bootstrap fanno riferimento all'elenco di broker che un client Apache Kafka può utilizzare per connettersi a un cluster Amazon MSK. Questo elenco potrebbe non includere tutti i broker del cluster. Puoi ottenere broker bootstrap utilizzando AWS Management Console AWS CLI, o l'API Amazon MSK.

Argomenti

- Ottieni i broker bootstrap usando il AWS Management Console
- Ottieni i broker bootstrap usando il AWS CLI

- Ottieni i broker bootstrap utilizzando l'API

Ottieni i broker bootstrap usando il AWS Management Console

Questo processo descrive come ottenere broker bootstrap per un cluster utilizzando il. AWS Management Console Il termine broker di bootstrap si riferisce a un elenco di broker che un client Apache Kafka può utilizzare come punto di partenza per connettersi al cluster. Questo elenco non include necessariamente tutti i broker di un cluster.

1. Accedere a e aprire la console Amazon MSK da https://console.aws.amazon.com/msk/casa? AWS Management Console region=us-east-1#/home/.
2. La tabella mostra tutti i cluster per la regione corrente in questo account. Scegli il nome di un cluster per visualizzarne la descrizione.
3. Nella pagina Riepilogo del cluster, scegli Visualizza informazioni sul client. Questo mostra i broker bootstrap e la stringa di connessione Apache. ZooKeeper

Ottieni i broker bootstrap usando il AWS CLI

Esegui il comando seguente, sostituendolo *ClusterArn* con l'Amazon Resource Name (ARN) che hai ottenuto quando hai creato il cluster. Se non disponi dell'ARN per il cluster, puoi trovarlo elencando tutti i cluster. Per ulteriori informazioni, consulta the section called “Elenca i cluster”.

```
aws kafka get-bootstrap-brokers --cluster-arn ClusterArn
```

Per un cluster MSK che utilizza the section called “Controllo degli accessi IAM”, l'output di questo comando è simile all'esempio JSON seguente.

```
{
    "BootstrapBrokerStringSaslIam": "b-1.myTestCluster.123z8u.c2.kafka.us-
west-1.amazonaws.com:9098,b-2.myTestCluster.123z8u.c2.kafka.us-
west-1.amazonaws.com:9098"
}
```

L'esempio seguente mostra i broker di bootstrap per un cluster con accesso pubblico attivato. Usa il `BootstrapBrokerStringPublicSaslIam` per l'accesso pubblico e la `BootstrapBrokerStringSaslIam` stringa per l'accesso dall'interno AWS.

```
{
```

```
    "BootstrapBrokerStringPublicSaslIam": "b-2-public.myTestCluster.v4ni96.c2.kafka-
beta.us-east-1.amazonaws.com:9198,b-1-public.myTestCluster.v4ni96.c2.kafka-
beta.us-east-1.amazonaws.com:9198,b-3-public.myTestCluster.v4ni96.c2.kafka-beta.us-
east-1.amazonaws.com:9198",
    "BootstrapBrokerStringSaslIam": "b-2.myTestCluster.v4ni96.c2.kafka-
beta.us-east-1.amazonaws.com:9098,b-1.myTestCluster.v4ni96.c2.kafka-beta.us-
east-1.amazonaws.com:9098,b-3.myTestCluster.v4ni96.c2.kafka-beta.us-
east-1.amazonaws.com:9098"
}
```

La stringa dei broker di bootstrap deve contenere tre broker provenienti da tutte le zone di disponibilità in cui è implementato il cluster MSK (a meno che non siano disponibili solo due broker).

Ottieni i broker bootstrap utilizzando l'API

Per fare in modo che i broker bootstrap utilizzino l'API, vedi. GetBootstrapBrokers

Monitora un cluster Amazon MSK Provisioned

Esistono diversi modi in cui Amazon MSK ti aiuta a monitorare lo stato del tuo cluster Amazon MSK Provisioned.

- Amazon MSK raccoglie i parametri di Apache Kafka e li invia ad Amazon CloudWatch dove puoi visualizzarli. Per ulteriori informazioni sui parametri Apache Kafka, inclusi quelli esposti da Amazon MSK, consulta la pagina Monitoring nella documentazione di Apache Kafka.
- Puoi anche monitorare il cluster MSK con Prometheus, un'applicazione di monitoraggio open source. Per informazioni su Prometheus, consulta la sezione relativa alla panoramica nella documentazione di Prometheus. Per informazioni su come monitorare il cluster MSK Provisioned con Prometheus, vedere. the section called “Monitora con Prometheus”
- (Solo broker standard) Amazon MSK ti aiuta a monitorare la capacità di storage su disco inviandoti automaticamente avvisi sulla capacità di storage quando un cluster Provisioned sta per raggiungere il limite di capacità di storage. Gli avvisi forniscono anche raccomandazioni sulle misure migliori da intraprendere per risolvere i problemi rilevati. Ciò consente di identificare e risolvere rapidamente i problemi relativi alla capacità del disco prima che diventino critici. Amazon MSK invia automaticamente questi avvisi alla console Amazon MSK, ad AWS Health Dashboard Amazon EventBridge e ai contatti e-mail del tuo account. AWS Per ulteriori informazioni sull'aumento della capacità di archiviazione, consulta Usa gli avvisi sulla capacità di archiviazione di Amazon MSK.

Argomenti

Visualizza i parametri di Amazon MSK utilizzando CloudWatch

Puoi monitorare i parametri per Amazon MSK utilizzando la CloudWatch console, la riga di comando o l' CloudWatch API. Le procedure seguenti mostrano come accedere ai parametri utilizzando questi diversi metodi.

Per accedere alle metriche utilizzando la console CloudWatch

Accedi a AWS Management Console e apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

1. Nel riquadro di navigazione, seleziona Parametri.
2. Scegli la scheda Tutti i parametri, quindi scegli AWS/Kafka.
3. Per visualizzare i parametri a livello di argomento, scegliere Topic, Broker ID, Cluster Name (Argomento, ID broker, Nome cluster); per parametri a livello di broker, scegliere Broker ID, Cluster Name (ID broker, Nome cluster); e per parametri a livello di cluster, scegliere Cluster Name (Nome cluster).
4. (Facoltativo) Nel riquadro grafico, seleziona una statistica e un periodo di tempo, quindi crea un CloudWatch allarme utilizzando queste impostazioni.

Per accedere alle metriche utilizzando il AWS CLI

Utilizzate le metriche e i comandi dell'elenco. get-metric-statistics

Per accedere alle metriche utilizzando la CLI CloudWatch

Utilizza i comandi mon-list-metrics e mon-get-stats.

Per accedere alle metriche utilizzando l'API CloudWatch

Utilizzare le operazioni ListMetrics e GetMetricStatistics.

Metriche di Amazon MSK per il monitoraggio dei broker Standard con CloudWatch

Amazon MSK si integra con Amazon per CloudWatch consentirti di raccogliere, visualizzare e analizzare i CloudWatch parametri per i tuoi broker MSK Standard. Le metriche configurate per i cluster MSK Provisioned vengono raccolte automaticamente e inserite a intervalli di 1 minuto. CloudWatch È possibile impostare il livello di monitoraggio per un cluster MSK Provisioned su uno dei seguenti:,, o. `DEFAULT` `PER_BROKER` `PER_TOPIC_PER_BROKER` `PER_TOPIC_PER_PARTITION` Le tabelle nelle sezioni seguenti mostrano tutti i parametri resi disponibili a partire da ciascun livello di monitoraggio.

> **Note**
>
> I nomi di alcuni parametri di Amazon MSK per il CloudWatch monitoraggio sono cambiati nella versione 3.6.0 e successive. Usa i nuovi nomi per monitorare questi parametri. Per i parametri con nomi modificati, la tabella seguente mostra il nome utilizzato nella versione 3.6.0 e successive, seguito dal nome nella versione 2.8.2.tiered.

I parametri del livello `DEFAULT` sono gratuiti. I prezzi per altre metriche sono descritti nella pagina CloudWatchdei prezzi di Amazon.

Monitoraggio del livello **`DEFAULT`**

I parametri descritti nella tabella seguente sono disponibili a livello di monitoraggio `DEFAULT` e sono gratuiti.

Nome	Quando visibile	Dimensi i	Descrizione
`ActiveController Count`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome del cluster	Solo un controller per cluster deve essere attivo in qualsiasi momento.
`BurstBalance`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il saldo residuo dei crediti di espansione input-output per i volumi EBS nel cluster. Utilizzalo per

Nome	Quando visibile	Dimensioni	Descrizione
			analizzare la latenza o la riduzione della velocità di trasmissione effettiva. `BurstBalance` non viene riportato per i volumi EBS quando le prestazioni di base di un volume sono maggiori delle prestazioni massime di espansione. Per ulteriori informazioni, consulta la pagina I/O Credits and burst performance.
`BytesInPerSec`	Dopo aver creato un argomento.	Nome cluster, ID broker, argomen	Il numero di byte al secondo ricevuti dai client. Questo parametro è disponibile per broker e anche per argomento.
`BytesOutPerSec`	Dopo aver creato un argomento.	Nome cluster, ID broker, argomen	Il numero di byte al secondo inviati ai client. Questo parametro è disponibile per broker e anche per argomento.
`ClientConnectionCount`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome del cluster, ID broker, autenticazione client	Il numero di connessioni client autenticate attive.

Nome	Quando visibile	Dimensioni	Descrizione
`ConnectionCount`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di connessioni attive autenticate, non autenticate e tra broker.
`CPUCreditBalance`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di crediti CPU ottenuti da un broker da quando è stato lanciato. I crediti vengono accumulati nel saldo del credito dopo che sono stati ottenuti e rimossi dal saldo del credito una volta spesi. L'esaurimento del credito della CPU può avere un impatto negativo sulle prestazioni del cluster. È possibile adottare delle misure per ridurre il carico della CPU. Ad esempio, puoi ridurre il numero di richieste dei client o aggiornare il tipo di broker a un tipo di broker M5.
`CpuIdle`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La percentuale di tempo di inattività della CPU.
`CpuIoWait`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La percentuale di inattività della CPU durante un'operazione su disco in sospeso.
`CpuSystem`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La percentuale di CPU nello spazio del kernel.

Nome	Quando visibile	Dimensioni	Descrizione
`CpuUser`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La percentuale di CPU nello spazio utente.
`GlobalPartitionCount`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome del cluster	Il numero di partizioni in tutti gli argomenti del cluster, escluse le repliche. Poiché `GlobalPartitionCount` non include le repliche, la somma dei `PartitionCount` valori può essere superiore a quella che si otterrebbe GlobalPartitionCount se il fattore di replica per un argomento fosse maggiore di 1.
`GlobalTopicCount`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome del cluster	Numero totale di argomenti in tutti i broker nel cluster.
`EstimatedMaxTimeLag`	Dopo che un gruppo di consumatori ha utilizzato un argomento.	Nome del cluster, gruppo di consumatori, argomento	Tempo stimato (in secondi) per lo svuotamento di `MaxOffsetLag` .
`KafkaAppLogsDiskUsed`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La percentuale di spazio su disco utilizzata per i log delle applicazioni.

Nome	Quando visibile	Dimensi i	Descrizione
`KafkaDataLogsDiskUsed` (dimensione `Cluster Name, Broker ID`)	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La percentuale di spazio su disco utilizzato per i log dei dati.
`LeaderCount`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero totale di leader delle partizioni per broker, escluse le repliche.
`MaxOffsetLag`	Dopo che un gruppo di consumatori ha utilizzato un argomento.	Nome del cluster, gruppo di consuma ri, argomen	Il ritardo massimo di offset su tutte le partizioni di un argomento.
`MemoryBuffered`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La dimensione in byte di memoria nel buffer per il broker.
`MemoryCached`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La dimensione in byte di memoria nella cache per il broker.
`MemoryFree`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La dimensione in byte di memoria libera e disponibile per il broker.

Nome	Quando visibile	Dimensioni	Descrizione
`HeapMemoryAfterGC`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La percentuale di memoria heap totale in uso dopo la rimozione di oggetti inutili.
`MemoryUsed`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La dimensione in byte di memoria utilizzata per il broker.
`MessagesInPerSec`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di messaggi in entrata al secondo per il broker.
`NetworkRxDropped`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di pacchetti ricezione eliminati.
`NetworkRxErrors`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di errori ricezione di rete per il broker.
`NetworkRxPackets`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di pacchetti ricevuti dal broker.
`NetworkTxDropped`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di pacchetti trasmissione eliminati.

Nome	Quando visibile	Dimensioni	Descrizione
`NetworkTxErrors`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di errori trasmissione di rete per il broker.
`NetworkTxPackets`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di pacchetti trasmessi dal broker.
`OfflinePartitionsCount`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome del cluster	Numero totale di partizioni che sono offline nel cluster.
`PartitionCount`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero totale di partizioni di argomento per broker, incluse le repliche.
`ProduceTotalTimeMsMean`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il tempo di produzione medio in millisecondi.
`RequestBytesMean`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero medio di byte della richiesta per il broker.
`RequestTime`	Dopo l'applicazione del throttling della richiesta.	Nome cluster, ID broker	Il tempo medio in millisecondi trascorso nella rete di broker e nei thread I/O per elaborare le richieste.

Nome	Quando visibile	Dimensi i	Descrizione
`RootDiskUsed`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La percentuale del disco radice utilizzato dal broker.
`SumOffsetLag`	Dopo che un gruppo di consumatori ha utilizzato un argomento.	Nome del cluster, gruppo di consuma ri, argomen	Il ritardo di offset aggregato per tutte le partizioni di un argomento.
`SwapFree`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La dimensione in byte della memoria swap disponibile per il broker.
`SwapUsed`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La dimensione in byte della memoria swap utilizzata dal broker.
`TrafficShaping`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Parametri di alto livello che indicano il numero di pacchetti formati (abbandonati o messi in coda) a causa di un eccesso di allocazioni di rete. Maggiori dettagli sono disponibili con i parametri PER_BROKER.
`UnderMinI srPartiti onCount`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di partizioni minIsr under per il broker.

Nome	Quando visibile	Dimensi i	Descrizione
`UnderReplicatedPartitions`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di partizioni replicate per il broker.
`UserPartitionExists`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Una metrica booleana che indica la presenza di una partizione di proprietà dell'utente su un broker. Il valore 1 indica la presenza di partizioni sul broker.
`ZooKeeperRequestLatencyMsMean`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Per cluster ZooKeeper basato. La latenza media in millisecondi per le richieste ZooKeeper Apache al broker.
`ZooKeeperSessionState`	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Per cluster basato. ZooKeeper Stato della connessione della ZooKeeper sessione del broker che può essere una delle seguenti: NOT_CONNECTED: '0.0', ASSOCIATING: '0.1', CONNECTING: '0.5', CONNECTEDREADONLY: '0.8', CONNECTED: '1.0', CLOSED: '5.0', AUTH_FAILED: '10.0'.

Monitoraggio del livello **`PER_BROKER`**

Quando imposti il livello di monitoraggio su `PER_BROKER`, ottieni i parametri descritti nella tabella seguente oltre a tutti i parametri del livello `DEFAULT`. Paghi per i parametri nella tabella seguente, mentre i parametri del livello `DEFAULT` continuano a essere gratuiti. I parametri contenuti in questa tabella hanno le seguenti dimensioni: Nome cluster, ID broker.

Nome	Quando visibile	Descrizione
`BwInAllowanceExceeded`	Dopo che il cluster raggiunge lo stato ACTIVE.	Numero di pacchetti modellati perché la larghezza di banda aggregata in entrata ha superato il valore massimo per l'istanza.
`BwOutAllowanceExceeded`	Dopo che il cluster raggiunge lo stato ACTIVE.	Numero di pacchetti modellati perché la larghezza di banda aggregata in uscita ha superato il valore massimo per l'istanza.
`ConntrackAllowanceExceeded`	Dopo che il cluster raggiunge lo stato ACTIVE.	Numero di pacchetti modellati perché il tracciamento della connessione ha superato il valore massimo per il broker. Il tracciamento della connessione è legato ai gruppi di sicurezza che tengono traccia di ogni connessione stabilita per garantire che i pacchetti restituiti vengano consegnati come previsto.
`ConnectionCloseRate`	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero di connessioni chiuse al secondo per ascoltatore. Questo numero viene aggregato per ascoltatore e filtrato per gli ascoltatori client.
`ConnectionCreationRate`	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero di nuove connessioni stabilite al secondo per ascoltatore. Questo numero viene aggregato per ascoltatore e filtrato per gli ascoltatori client.
`CpuCreditUsage`	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero di crediti CPU utilizzati dal broker. L'esaurimento del credito della CPU può avere un impatto negativo sulle prestazioni del cluster. È possibile adottare delle misure

Nome	Quando visibile	Descrizione
		per ridurre il carico della CPU. Ad esempio, puoi ridurre il numero di richieste dei client o aggiornare il tipo di broker a un tipo di broker M5.
`FetchConsumerLocalTimeMsMean`	Dopo che c'è un produttore/consumatore.	Tempo medio in millisecondi di elaborazione della richiesta del consumatore presso il leader.
`FetchConsumerRequestQueueTimeMsMean`	Dopo che c'è un produttore/consumatore.	Tempo medio in millisecondi di attesa della richiesta del consumatore nella coda delle richieste.
`FetchConsumerResponseQueueTimeMsMean`	Dopo che c'è un produttore/consumatore.	Tempo medio in millisecondi di attesa della richiesta del consumatore nella coda delle risposte.
`FetchConsumerResponseSendTimeMsMean`	Dopo che c'è un produttore/consumatore.	Tempo medio in millisecondi impiegato dal consumatore per inviare una risposta.
`FetchConsumerTotalTimeMsMean`	Dopo che c'è un produttore/consumatore.	Il tempo totale medio in millisecondi impiegato dai consumatori per recuperare i dati dal broker.
`FetchFollowerLocalTimeMsMean`	Dopo che c'è un produttore/consumatore.	Tempo medio in millisecondi impiegato a livello di leader per elaborare la richiesta follower.
`FetchFollowerRequestQueueTimeMsMean`	Dopo che c'è un produttore/consumatore.	Tempo medio in millisecondi di attesa della richiesta follower nella coda delle richieste.
`FetchFollowerResponseQueueTimeMsMean`	Dopo che c'è un produttore/consumatore.	Tempo medio in millisecondi di attesa della richiesta follower nella coda delle risposte.

Nome	Quando visibile	Descrizione
FetchFollowerResponseSendTimeMsMean	Dopo che c'è un produttore/consumatore.	Tempo medio in millisecondi impiegato dal follower per inviare una risposta.
FetchFollowerTotalTimeMsMean	Dopo che c'è un produttore/consumatore.	Il tempo totale medio in millisecondi impiegato dai follower per recuperare i dati dal broker.
FetchMessageConversionsPerSec	Dopo aver creato un argomento.	Il numero di conversioni dei messaggi di recupero al secondo per il broker.
FetchThrottleByteRate	Dopo l'applicazione del throttling della larghezza di banda.	Il numero di byte sottoposti a throttling al secondo.
FetchThrottleQueueSize	Dopo l'applicazione del throttling della larghezza di banda.	Il numero di messaggi nella coda di throttling.
FetchThrottleTime	Dopo l'applicazione del throttling della larghezza di banda.	Il tempo medio del throttling di recupero in millisecondi.
IAMNumberOfConnectionRequests	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero di richieste di autenticazione IAM al secondo.
IAMTooManyConnections	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero di connessioni tentate è superiore a 100. 0 indica che il numero di connessioni rientra nel limite. Se >0, il limite di accelerazione viene superato ed è necessario ridurre il numero di connessioni.
NetworkProcessorAvgIdlePercent	Dopo che il cluster raggiunge lo stato ACTIVE.	La percentuale media del tempo di inattività dei processori di rete.

Nome	Quando visibile	Descrizione
PpsAllowanceExceeded	Dopo che il cluster raggiunge lo stato ACTIVE.	Numero di pacchetti modellati perché il PPS bidirezionale ha superato il valore massimo per il broker.
ProduceLocalTimeMsMean	Dopo che il cluster raggiunge lo stato ACTIVE.	Tempo medio in millisecondi impiegato a livello di leader per elaborare la richiesta.
ProduceMessageConversionsPerSec	Dopo aver creato un argomento.	Il numero di conversioni di messaggi di produzione al secondo per il broker.
ProduceMessageConversionsTimeMsMean	Dopo che il cluster raggiunge lo stato ACTIVE.	Il tempo medio in millisecondi impiegato per le conversioni di formato dei messaggi.
ProduceRequestQueueTimeMsMean	Dopo che il cluster raggiunge lo stato ACTIVE.	Il tempo medio in millisecondi che i messaggi di richiesta rimangono nella coda.
ProduceResponseQueueTimeMsMean	Dopo che il cluster raggiunge lo stato ACTIVE.	Il tempo medio in millisecondi che messaggi di risposta rimangono nella coda.
ProduceResponseSendTimeMsMean	Dopo che il cluster raggiunge lo stato ACTIVE.	Il tempo medio in millisecondi impiegato per l'invio di messaggi di risposta.
ProduceThrottleByteRate	Dopo l'applicazione del throttling della larghezza di banda.	Il numero di byte sottoposti a throttling al secondo.
ProduceThrottleQueueSize	Dopo l'applicazione del throttling della larghezza di banda.	Il numero di messaggi nella coda di throttling.

Nome	Quando visibile	Descrizione
`ProduceThrottleTime`	Dopo l'applicazione del throttling della larghezza di banda.	Il tempo di throttling di produzione medio in millisecondi.
`ProduceTotalTimeMs Mean`	Dopo che il cluster raggiunge lo stato ACTIVE.	Il tempo di produzione medio in millisecondi.
`RemoteFetchBytesPe rSec (RemoteBy tesInPerSec in v2.8.2.tiered)`	Dopo che è presente un produttore/ consumatore.	Il numero totale di byte trasferiti dall'archiviazione a più livelli in risposta alle richieste dei consumatori. Questo parametro include tutte le partizion i di argomento che contribuiscono al traffico di trasferimento dati a valle. Categoria: traffico e tassi di errore. Questo è un parametro KIP-405.
`RemoteCopyBytesPerSec (RemoteBytesOutPerSec in v2.8.2.tiered)`	Dopo che è presente un produttore/ consumatore.	Il numero totale di byte trasferiti nell'archiviazione a più livelli, inclusi i dati provenienti da segmenti di log, indici e altri file ausiliari. Questo parametro include tutte le partizion i di argomento che contribuiscono al traffico di trasferimento dati a monte. Categoria: traffico e tassi di errore. Questo è un parametro KIP-405.
`RemoteLogManagerTa sksAvgIdlePercent`	Dopo che il cluster raggiunge lo stato ACTIVE.	La percentuale media di tempo che il gestore di log remoto ha trascorso inattivo. Il gestore remoto dei log trasferisce i dati dal broker all'archi viazione a più livelli. Categoria: attività interna. Questo è un parametro KIP-405.

Nome	Quando visibile	Descrizione
`RemoteLogReaderAvgIdlePercent`	Dopo che il cluster raggiunge lo stato ACTIVE.	La percentuale media di tempo che il lettore di log remoto ha trascorso inattivo. Il lettore di log remoto trasferisce i dati dall'archiviazione remota al broker in risposta alle richieste dei consumatori. Categoria: attività interna. Questo è un parametro KIP-405.
`RemoteLogReaderTaskQueueSize`	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero di attività responsabili delle letture dall'archiviazione a più livelli in attesa di essere pianificate. Categoria: attività interna. Questo è un parametro KIP-405.
`RemoteFetchErrorsPerSec (RemoteReadErrorPerSec in v2.8.2.tiered)`	Dopo che il cluster raggiunge lo stato ACTIVE.	La percentuale totale di errori in risposta alle richieste di lettura che il broker specificato ha inviato all'archiviazione a più livelli per recuperare i dati in risposta alle richieste dei consumatori. Questo parametro include tutte le partizioni di argomento che contribuiscono al traffico di trasferimento dati a valle. Categoria: traffico e tassi di errore. Questo è un parametro KIP-405.

Nome	Quando visibile	Descrizione
`RemoteFetchRequestsPerSec (RemoteReadRequestsPerSec in v2.8.2.tiered)`	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero totale di richieste di lettura che il broker specificato ha inviato all'archiviazione a più livelli per recuperare i dati in risposta alle richieste dei consumatori. Questo parametro include tutte le partizioni di argomento che contribuiscono al traffico di trasferimento dati a valle. Categoria: traffico e tassi di errore. Questo è un parametro KIP-405.
`RemoteCopyErrorsPerSec (RemoteWriteErrorPerSec in v2.8.2.tiered)`	Dopo che il cluster raggiunge lo stato ACTIVE.	La percentuale totale di errori in risposta alle richieste di scrittura che il broker specificato ha inviato all'archiviazione a più livelli per trasferire i dati a monte. Questo parametro include tutte le partizioni di argomento che contribuiscono al traffico di trasferimento dati a monte. Categoria: traffico e tassi di errore. Questo è un parametro KIP-405.
`RemoteLogSizeBytes`	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero di byte archiviati sul livello remoto. Questa metrica è disponibile per i cluster di storage su più livelli di Apache Kafka versione 3.7.x su Amazon MSK.
`ReplicationBytesInPerSec`	Dopo aver creato un argomento.	Il numero di byte al secondo ricevuti da altri broker.
`ReplicationBytesOutPerSec`	Dopo aver creato un argomento.	Il numero di byte al secondo inviati ad altri broker.

Nome	Quando visibile	Descrizione
`RequestExemptFromThrottleTime`	Dopo l'applicazione del throttling della richiesta.	Il tempo medio in millisecondi trascorso nella rete di broker e nei thread I/O per elaborare le richieste esenti da throttling.
`RequestHandlerAvgIdlePercent`	Dopo che il cluster raggiunge lo stato ACTIVE.	La percentuale media del tempo di inattività dei thread del gestore di richieste.
`RequestThrottleQueueSize`	Dopo l'applicazione del throttling della richiesta.	Il numero di messaggi nella coda di throttling.
`RequestThrottleTime`	Dopo l'applicazione del throttling della richiesta.	Il tempo di throttling della richiesta medio in millisecondi.
`TcpConnections`	Dopo che il cluster raggiunge lo stato ACTIVE.	Mostra il numero di segmenti TCP in entrata e in uscita con il flag SYN impostato.
`RemoteCopyLagBytes (TotalTierBytesLag in v2.8.2.tiered)`	Dopo aver creato un argomento.	Il numero totale di byte dei dati idonei per l'archiviazione a più livelli sul broker ma che non sono ancora stati trasferiti in tale archiviazione. Questi parametri mostrano l'efficienza del trasferimento dati a monte. Con l'aumentare del ritardo, aumenta la quantità di dati che non persistono nell'archiviazione a più livelli. Categoria : ritardo di archiviazione. Questo non è un parametro KIP-405.

Nome	Quando visibile	Descrizione
`TrafficBytes`	Dopo che il cluster raggiunge lo stato ACTIVE.	Mostra il traffico di rete in byte complessivi tra client (produttori e consumatori) e broker. Il traffico tra i broker non viene segnalato.
`VolumeQueueLength`	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero di richieste di operazioni di lettura e scrittura in attesa di completamento nel periodo di tempo specificato.
`VolumeReadBytes`	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero di byte letti durante il periodo di tempo specificato.
`VolumeReadOps`	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero totale di operazioni di lettura nel periodo di tempo specificato.
`VolumeTotalReadTime`	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero totale di secondi impiegato da tutte le operazioni di lettura completate nel periodo di tempo specificato.
`VolumeTotalWriteTime`	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero totale di secondi impiegato da tutte le operazioni di scrittura completate nel periodo di tempo specificato.
`VolumeWriteBytes`	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero di byte scritti durante il periodo di tempo specificato.
`VolumeWriteOps`	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero totale di operazioni di scrittura durante il periodo di tempo specificato.

Monitoraggio del livello **PER_TOPIC_PER_BROKER**

Quando imposti il livello di monitoraggio su PER_TOPIC_PER_BROKER, ottieni i parametri descritti nella tabella seguente, oltre a tutti i parametri dei livelli PER_BROKER e DEFAULT. Solo i parametri del livello DEFAULT sono gratuiti. I parametri contenuti in questa tabella hanno le seguenti dimensioni: Nome cluster, ID broker, Argomento.

> **Important**
>
> Per un cluster Amazon MSK che utilizza Apache Kafka 2.4.1 o una versione più recente, i parametri nella tabella seguente vengono visualizzati solo dopo che i loro valori diventano diversi da zero per la prima volta. Ad esempio, per visualizzare BytesInPerSec, uno o più produttori devono prima inviare i dati al cluster.

Nome	Quando visibile	Descrizione
FetchMessageConversionsPerSec	Dopo aver creato un argomento.	Il numero di messaggi recuperati convertiti al secondo.
MessagesInPerSec	Dopo aver creato un argomento.	Il numero di messaggi ricevuti al secondo.
ProduceMessageConversionsPerSec	Dopo aver creato un argomento.	Il numero di conversioni al secondo per i messaggi prodotti.
RemoteFetchBytesPerSec (RemoteBytesInPerSec in v2.8.2.tiered)	Dopo aver creato un argomento, l'argomento è in fase di produzione/utilizzo.	Il numero di byte trasferiti nell'archiviazione a più livelli in risposta alle richieste dei consumatori per l'argomento e il broker specificati. Questo parametro include tutte le partizioni dell'argomento che contribuiscono al traffico di trasferimento dati a valle sul broker specificato. Categoria: traffico e tassi di errore. Questo è un parametro KIP-405.

Nome	Quando visibile	Descrizione
`RemoteCopyBytesPerSec (RemoteBytesOutPerSec in v2.8.2.tiered)`	Dopo aver creato un argomento, l'argomento è in fase di produzione/utilizzo.	Il numero di byte trasferiti nell'archiviazione a più livelli per l'argomento e il broker specificati. Questo parametro include tutte le partizioni dell'argomento che contribuiscono al traffico di trasferimento dati a monte sul broker specificato. Categoria: traffico e tassi di errore. Questo è un parametro KIP-405.
`RemoteFetchErrorsPerSec (RemoteReadErrorPerSec in v2.8.2.tiered)`	Dopo aver creato un argomento, l'argomento è in fase di produzione/utilizzo.	La percentuale di errori in risposta alle richieste di lettura che il broker specificato invia all'archiviazione a più livelli per recuperare i dati in risposta alle richieste dei consumatori in relazione all'argomento specificato. Questo parametro include tutte le partizioni dell'argomento che contribuiscono al traffico di trasferimento dati a valle sul broker specificato. Categoria: traffico e tassi di errore. Questo è un parametro KIP-405.
`RemoteFetchRequestsPerSec (RemoteReadRequestsPerSec in v2.8.2.tiered)`	Dopo aver creato un argomento, l'argomento è in fase di produzione/utilizzo.	Il numero di richieste di lettura che il broker specificato invia all'archiviazione a più livelli per recuperare i dati in risposta alle richieste dei consumatori in relazione all'argomento specificato. Questo parametro include tutte le partizioni dell'argomento che contribuiscono al traffico di trasferimento dati a valle sul broker specificato. Categoria: traffico e tassi di errore. Questo è un parametro KIP-405.

Nome	Quando visibile	Descrizione
`RemoteCopyErrorsPerSec (RemoteWriteErrorPerSec in v2.8.2.tiered)`	Dopo aver creato un argomento, l'argomento è in fase di produzione/utilizzo.	La percentuale di errori in risposta alle richieste di scrittura che il broker specificato invia all'archiviazione a più livelli per trasferire i dati a monte. Questo parametro include tutte le partizioni dell'argomento che contribuiscono al traffico di trasferimento dati a monte sul broker specificato. Categoria: traffico e tassi di errore. Questo è un parametro KIP-405.
`RemoteLogSizeBytes`	Dopo aver creato un argomento.	Il numero di byte archiviati sul livello remoto. Questa metrica è disponibile per i cluster di storage su più livelli di Apache Kafka versione 3.7.x su Amazon MSK.

Monitoraggio del livello **`PER_TOPIC_PER_PARTITION`**

Quando imposti il livello di monitoraggio su `PER_TOPIC_PER_PARTITION`, ottieni i parametri descritti nella tabella seguente, oltre a tutti i parametri dei livelli `PER_TOPIC_PER_BROKER`, `PER_BROKER` e DEFAULT. Solo i parametri del livello `DEFAULT` sono gratuiti. I parametri in questa tabella hanno le seguenti dimensioni: gruppo di consumatori, argomento, partizione.

Nome	Quando visibile	Descrizione
`EstimatedTimeLag`	Dopo che un gruppo di consumatori ha utilizzato un argomento.	Tempo stimato (in secondi) per eliminare il ritardo di offset della partizione.
`OffsetLag`	Dopo che un gruppo di consumatori ha utilizzato un argomento.	Ritardo del consumatore a livello di partizione nel numero di offset.

Comprendi gli stati del cluster MSK Provisioned

La tabella seguente mostra i possibili stati di un cluster MSK Provisioned e ne descrive il significato. Se non diversamente specificato, gli stati del cluster MSK Provisioned si applicano sia ai tipi di broker Standard che Express. Questa tabella descrive anche le azioni che è possibile e non è possibile eseguire quando un cluster MSK Provisioned si trova in uno di questi stati. Per scoprire lo stato di un cluster, consulta la AWS Management Console. È inoltre possibile utilizzare il comando describe-cluster-v2 o l'operazione DescribeClusterV2 per descrivere il cluster Provisioned. La descrizione di un cluster include il relativo stato.

Stato del cluster MSK Provisioned	Significato e operazioni possibili
ACTIVE	Puoi produrre e utilizzare dati. Puoi anche eseguire l'API e AWS CLI le operazioni di Amazon MSK sul cluster.
CREAZIONE IN CORSO	Amazon MSK sta configurando il cluster Provisioned. È necessario attendere che il cluster raggiunga lo stato ATTIVO prima di poterlo utilizzare per produrre o consumare dati o per eseguire l'API o AWS CLI le operazioni di Amazon MSK su di esso.
ELIMINAZIONE IN CORSO	Il cluster Provisioned viene eliminato. Non è possibile utilizzarlo per produrre o utilizzare dati. Inoltre, non è possibile eseguire l'API o AWS CLI le operazioni di Amazon MSK su di essa.
Non riuscito	Il processo di creazione o eliminazione del cluster Provisioned non è riuscito. Non è possibile utilizzare il cluster per produrre o utilizzare dati. Puoi eliminare il cluster ma non puoi eseguire operazioni di API Amazon MSK o AWS CLI aggiornarlo.
HEALING	Amazon MSK sta eseguendo un'operazione interna, ad esempio la sostituzione di un

Stato del cluster MSK Provisioned	Significato e operazioni possibili
	broker non funzionante. Ad esempio, il broker potrebbe non rispondere. Puoi comunque utilizzare il cluster Provisioned per produrre e consumare dati. Tuttavia, non puoi eseguire l'API Amazon MSK o AWS CLI aggiornare le operazioni sul cluster finché non torna allo stato ACTIVE.
MAINTENANCE	(Solo broker standard) Amazon MSK esegue operazioni di manutenzione ordinaria sul cluster. Tali operazioni di manutenzione includono l'applicazione di patch di sicurezza. È ancora possibile utilizzare il cluster per produrre e utilizzare dati. Tuttavia, non è possibile eseguire operazioni di aggiornamento dell'API o della AWS CLI di Amazon MSK sul cluster finché non torna allo stato ACTIVE. Lo stato del cluster rimane ATTIVO durante la manutenzi one sui broker Express. Consultare Rattoppare.
REBOOTING_BROKER	Amazon MSK sta riavviando un broker. È ancora possibile utilizzare il cluster Provision ed per produrre e consumare dati. Tuttavia, non puoi eseguire l'API Amazon MSK o AWS CLI aggiornare le operazioni sul cluster finché non torna allo stato ACTIVE.
AGGIORNAMENTO IN CORSO	Un'API o un' AWS CLI operazione Amazon MSK avviata dall'utente sta aggiornando il cluster Provisioned. Puoi comunque utilizzare il cluster Provisioned per produrre e consumare dati. Tuttavia, non è possibile eseguire alcuna API Amazon MSK aggiuntiva o eseguire operazioni di AWS CLI aggiornamento sul cluster finché non torna allo stato ATTIVO.

Metriche di Amazon MSK per il monitoraggio dei broker Express con CloudWatch

Amazon MSK si integra per CloudWatch consentirti di raccogliere, visualizzare e analizzare i CloudWatch parametri per i tuoi broker MSK Express. Le metriche configurate per i cluster MSK Provisioned vengono raccolte automaticamente e inserite a intervalli di 1 minuto. CloudWatch È possibile impostare il livello di monitoraggio per un cluster MSK Provisioned su uno dei seguenti:,, o. `DEFAULT` `PER_BROKER` `PER_TOPIC_PER_BROKER` `PER_TOPIC_PER_PARTITION` Le tabelle nelle sezioni seguenti mostrano le metriche disponibili a partire da ogni livello di monitoraggio.

I parametri del livello `DEFAULT` sono gratuiti. I prezzi per altre metriche sono descritti nella pagina CloudWatchdei prezzi di Amazon.

`DEFAULT`Monitoraggio dei livelli per i broker Express

Le metriche descritte nella tabella seguente sono disponibili gratuitamente a livello di `DEFAULT` monitoraggio.

Nome	Quando visibile	Dimensioni	Descrizione
ActiveControllerCount	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome del cluster	Solo un controller per cluster deve essere attivo in qualsiasi momento.
BytesInPerSec	Dopo aver creato un argomento.	Nome cluster, ID broker, argomento	Il numero di byte al secondo ricevuti dai client. Questo parametro è disponibile per broker e anche per argomento.
BytesOutPerSec	Dopo aver creato un argomento.	Nome cluster, ID broker, argomento	Il numero di byte al secondo inviati ai client. Questo parametro è disponibile per broker e anche per argomento.

Nome	Quando visibile	Dimensioni	Descrizione
ClientConnectionCount	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome del cluster, ID broker, autenticazione client	Il numero di connessioni client autenticate attive.
ConnectionCount	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di connessioni attive autenticate, non autenticate e tra broker.
CpuIdle	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La percentuale di tempo di inattività della CPU.
CpuSystem	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La percentuale di CPU nello spazio del kernel.
CpuUser	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La percentuale di CPU nello spazio utente.

Nome	Quando visibile	Dimensioni	Descrizione
GlobalPartitionCount	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome del cluster	Il numero di partizioni in tutti gli argomenti del cluster, escluse le repliche. Poiché `GlobalPartitionCount` non include le repliche, la somma dei `PartitionCount` valori può essere superiore a quella che si otterrebbe `GlobalPartitionCount` se il fattore di replica per un argomento fosse maggiore di. 1
GlobalTopicCount	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome del cluster	Numero totale di argomenti in tutti i broker nel cluster.
EstimatedMaxTimeLag	Dopo che un gruppo di consumatori ha utilizzato un argomento.	Gruppo di consumatori, argomento	Tempo stimato (in secondi) per lo svuotamento di `MaxOffsetLag` .
LeaderCount	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero totale di leader delle partizioni per broker, escluse le repliche.

Nome	Quando visibile	Dimensioni	Descrizione
MaxOffsetLag	Dopo che un gruppo di consumatori ha utilizzato un argomento.	Gruppo di consumatori, argomento	Il ritardo massimo di offset su tutte le partizioni di un argomento.
MemoryBuffered	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La dimensione in byte di memoria nel buffer per il broker.
MemoryCached	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La dimensione in byte di memoria nella cache per il broker.
MemoryFree	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La dimensione in byte di memoria libera e disponibile per il broker.
MemoryUsed	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	La dimensione in byte di memoria utilizzata per il broker.
MessagesInPerSec	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di messaggi in entrata al secondo per il broker.
NetworkRxDropped	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di pacchetti ricezione eliminati.
NetworkRxErrors	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di errori ricezione di rete per il broker.
NetworkRxPackets	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di pacchetti ricevuti dal broker.

Nome	Quando visibile	Dimensioni	Descrizione
NetworkTxDropped	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di pacchetti trasmissione eliminati.
NetworkTxErrors	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di errori trasmissione di rete per il broker.
NetworkTxPackets	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero di pacchetti trasmessi dal broker.
PartitionCount	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero totale di partizioni di argomento per broker, incluse le repliche.
ProduceTotalTimeMs Mean	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il tempo di produzione medio in millisecondi.
RequestBytesMean	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Il numero medio di byte della richiesta per il broker.
RequestTime	Dopo l'applicazione del throttling della richiesta.	Nome cluster, ID broker	Il tempo medio, in millisecondi, impiegato nella rete di broker e I/O nei thread per elaborare le richieste.
StorageUsed	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome del cluster	Lo storage totale utilizzato in tutte le partizioni del cluster, escluse le repliche.

Nome	Quando visibile	Dimensioni	Descrizione
SumOffsetLag	Dopo che un gruppo di consumatori ha utilizzato un argomento.	Gruppo di consumatori, argomento	Il ritardo di offset aggregato per tutte le partizioni di un argomento.
UserPartitionExists	Dopo che il cluster raggiunge lo stato ACTIVE.	Nome cluster, ID broker	Metrica booleana che indica la presenza di una partizione di proprietà dell'utente su un broker. Il valore 1 indica la presenza di partizioni sul broker.

`PER_BROKER`Monitoraggio del livello per i broker Express

Quando imposti il livello di monitoraggio su `PER_BROKER`, ottieni i parametri descritti nella tabella seguente oltre a tutti i parametri del livello `DEFAULT`. Paghi in base alle metriche riportate nella tabella seguente, mentre le metriche di `DEFAULT` livello continuano a essere gratuite. I parametri contenuti in questa tabella hanno le seguenti dimensioni: Nome cluster, ID broker.

Metriche aggiuntive disponibili a partire dal livello di monitoraggio PER_BROKER

Nome	Quando visibile	Descrizione
ConnectionCloseRate	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero di connessioni chiuse al secondo per ascoltatore. Questo numero viene aggregato per ascoltatore e filtrato per gli ascoltatori client.
ConnectionCreationRate	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero di nuove connessioni stabilite al secondo per ascoltatore. Questo numero viene aggregato per ascoltato

Nome	Quando visibile	Descrizione
		re e filtrato per gli ascoltatori client.
FetchConsumerLocalTimeMsMean	Dopo che c'è un produttore/consumatore.	Tempo medio in millisecondi di elaborazione della richiesta del consumatore presso il leader.
FetchConsumerRequestQueueTimeMsMean	Dopo che c'è un produttore/consumatore.	Tempo medio in millisecondi di attesa della richiesta del consumatore nella coda delle richieste.
FetchConsumerResponseQueueTimeMsMean	Dopo che c'è un produttore/consumatore.	Tempo medio in millisecondi di attesa della richiesta del consumatore nella coda delle risposte.
FetchConsumerResponseSendTimeMsMean	Dopo che c'è un produttore/consumatore.	Tempo medio in millisecondi impiegato dal consumatore per inviare una risposta.
FetchConsumerTotalTimeMsMean	Dopo che c'è un produttore/consumatore.	Il tempo totale medio in millisecondi impiegato dai consumatori per recuperare i dati dal broker.
FetchFollowerLocalTimeMsMean	Dopo che c'è un produttore/consumatore.	Tempo medio in millisecondi impiegato a livello di leader per elaborare la richiesta follower.
FetchFollowerRequestQueueTimeMsMean	Dopo che c'è un produttore/consumatore.	Tempo medio in millisecondi di attesa della richiesta follower nella coda delle richieste.

Nome	Quando visibile	Descrizione
FetchFollowerResponseQueueTimeMsMean	Dopo che c'è un produttore/consumatore.	Tempo medio in millisecondi di attesa della richiesta follower nella coda delle risposte.
FetchFollowerResponseSendTimeMsMean	Dopo che c'è un produttore/consumatore.	Tempo medio in millisecondi impiegato dal follower per inviare una risposta.
FetchFollowerTotalTimeMsMean	Dopo che c'è un produttore/consumatore.	Il tempo totale medio in millisecondi impiegato dai follower per recuperare i dati dal broker.
FetchThrottleByteRate	Dopo l'applicazione del throttling della larghezza di banda.	Il numero di byte sottoposti a throttling al secondo.
FetchThrottleQueueSize	Dopo l'applicazione del throttling della larghezza di banda.	Il numero di messaggi nella coda di throttling.
FetchThrottleTime	Dopo l'applicazione del throttling della larghezza di banda.	Il tempo medio del throttling di recupero in millisecondi.
IAMNumberOfConnectionRequests	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero di richieste di autenticazione IAM al secondo.

Nome	Quando visibile	Descrizione
IAMTooManyConnections	Dopo che il cluster raggiunge lo stato ACTIVE.	Il numero di connessioni tentate è superiore a 100. `0`significa che il numero di connessioni rientra nel limite. Se`>0`, il limite di acceleraz ione viene superato ed è necessario ridurre il numero di connessioni.
NetworkProcessorAvgIdlePerc ent	Dopo che il cluster raggiunge lo stato ACTIVE.	La percentuale media del tempo di inattività dei processori di rete.
ProduceLocalTimeMsMean	Dopo che il cluster raggiunge lo stato ACTIVE.	Tempo medio in millisecondi impiegato a livello di leader per elaborare la richiesta.
ProduceRequestQueu eTimeMsMean	Dopo che il cluster raggiunge lo stato ACTIVE.	Il tempo medio in millisecondi che i messaggi di richiesta rimangono nella coda.
ProduceResponseQue ueTimeMsMean	Dopo che il cluster raggiunge lo stato ACTIVE.	Il tempo medio in milliseco ndi che messaggi di risposta rimangono nella coda.
ProduceResponseSen dTimeMsMean	Dopo che il cluster raggiunge lo stato ACTIVE.	Il tempo medio in milliseco ndi impiegato per l'invio di messaggi di risposta.
ProduceThrottleByteRate	Dopo l'applicazione del throttling della larghezza di banda.	Il numero di byte sottoposti a throttling al secondo.
ProduceThrottleQueueSize	Dopo l'applicazione del throttling della larghezza di banda.	Il numero di messaggi nella coda di throttling.

Nome	Quando visibile	Descrizione
ProduceThrottleTime	Dopo l'applicazione del throttling della larghezza di banda.	Il tempo di throttling di produzione medio in millisecondi.
ProduceTotalTimeMsMean	Dopo che il cluster raggiunge lo stato ACTIVE.	Il tempo di produzione medio in millisecondi.
ReplicationBytesInPerSec	Dopo aver creato un argomento.	Il numero di byte al secondo ricevuti da altri broker.
ReplicationBytesOutPerSec	Dopo aver creato un argomento.	Il numero di byte al secondo inviati ad altri broker.
RequestExemptFromThrottleTime	Dopo l'applicazione del throttling della richiesta.	Il tempo medio, in millisecondi, impiegato nella rete di broker e nei I/O thread per elaborare le richieste esenti dal throttling.
RequestHandlerAvgIdlePercent	Dopo che il cluster raggiunge lo stato ACTIVE.	La percentuale media del tempo di inattività dei thread del gestore di richieste.
RequestThrottleQueueSize	Dopo l'applicazione del throttling della richiesta.	Il numero di messaggi nella coda di throttling.
RequestThrottleTime	Dopo l'applicazione del throttling della richiesta.	Il tempo di throttling della richiesta medio in millisecondi.
TcpConnections	Dopo che il cluster raggiunge lo stato ACTIVE.	Mostra il numero di segmenti TCP in entrata e in uscita con il flag SYN impostato.

Nome	Quando visibile	Descrizione
TrafficBytes	Dopo che il cluster raggiunge lo stato ACTIVE.	Mostra il traffico di rete in byte complessivi tra client (produttori e consumatori) e broker. Il traffico tra i broker non viene segnalato.

`PER_TOPIC_PER_PARTITION`monitoraggio del livello per i broker Express

Quando si imposta il livello di monitoraggio su`PER_TOPIC_PER_PARTITION`, si ottengono le metriche descritte nella tabella seguente, oltre a tutte le metriche dei livelli `PER_TOPIC_PER_BROKER`PER_BROKER, e. `DEFAULT` Solo le metriche `DEFAULT` di livello sono gratuite. I parametri in questa tabella hanno le seguenti dimensioni: gruppo di consumatori, argomento, partizione.

Metriche aggiuntive disponibili a partire dal livello di monitoraggio PER_PARTITION

Nome	Quando visibile	Descrizione
EstimatedTimeLag	Dopo che un gruppo di consumatori ha utilizzato un argomento.	Tempo stimato (in secondi) per eliminare il ritardo di offset della partizione.
OffsetLag	Dopo che un gruppo di consumatori ha utilizzato un argomento.	Ritardo del consumatore a livello di partizione nel numero di offset.

`PER_TOPIC_PER_BROKER`monitoraggio del livello per i broker Express

Quando si imposta il livello di monitoraggio su`PER_TOPIC_PER_BROKER`, si ottengono le metriche descritte nella tabella seguente, oltre a tutte le metriche dei `PER_BROKER` livelli and. `DEFAULT` Solo le metriche `DEFAULT` di livello sono gratuite. I parametri contenuti in questa tabella hanno le seguenti dimensioni: Nome cluster, ID broker, Argomento.

> **Important**
>
> Le metriche nella tabella seguente vengono visualizzate solo dopo che i relativi valori diventano diversi da zero per la prima volta. Ad esempio, per visualizzare BytesInPerSec, uno o più produttori devono prima inviare dati al cluster.

Metriche aggiuntive disponibili a partire dal livello di monitoraggio PER_TOPIC_PER_BROKER

Nome	Quando visibile	Descrizione
MessagesInPerSec	Dopo aver creato un argomento.	Il numero di messaggi ricevuti al secondo.

Monitora un cluster MSK Provisioned con Prometheus

È possibile monitorare il cluster MSK Provisioned con Prometheus, un sistema di monitoraggio open source per dati metrici di serie temporali. Puoi pubblicare questi dati su Servizio gestito da Amazon per Prometheus utilizzando la funzione di scrittura remota di Prometheus. Puoi anche utilizzare strumenti compatibili con i parametri in formato Prometheus o strumenti che si integrano con Amazon MSK Open Monitoring, come Datadog, Lenses, New Relic e Sumo logic. Il monitoraggio aperto è disponibile gratuitamente, ma per il trasferimento dei dati tra le zone di disponibilità vengono addebitati dei costi.

Per informazioni su Prometheus, consulta la documentazione di Prometheus.

Per informazioni sull'uso di Prometheus, consulta Migliora le informazioni operative per Amazon MSK usando Amazon Managed Service for Prometheus e Amazon Managed Grafana.

> **Note**
>
> KRaft la modalità metadati e i broker MSK Express non possono avere entrambi abilitati il monitoraggio aperto e l'accesso pubblico.

Abilita il monitoraggio aperto sui nuovi cluster MSK Provisioned

Questa procedura descrive come abilitare il monitoraggio aperto su un nuovo cluster MSK utilizzando AWS CLI l' AWS Management Console API Amazon MSK.

Usando il AWS Management Console

1. Accedere a e aprire la console Amazon MSK da https://console.aws.amazon.com/msk/casa?AWS Management Console region=us-east-1#/home/.
2. Nella sezione Monitoring (Monitoraggio), selezionare la casella di controllo accanto a Enable open monitoring with Prometheus (Abilita monitoraggio aperto con Prometheus).
3. Fornire le informazioni richieste in tutte le sezioni della pagina e rivedere tutte le opzioni disponibili.
4. Scegli Create cluster (Crea cluster).

Usando il AWS CLI

- Richiamare il comando create-cluster e specificarne l'opzione `open-monitoring`. Abilitare `JmxExporter`, `NodeExporter` o entrambi. Se si specifica `open-monitoring`, non è possibile disabilitare i due esportatori contemporaneamente.

Utilizzo dell'API

- Richiama l'CreateClusteroperazione e specifica`OpenMonitoring`. Abilitare `jmxExporter`, `nodeExporter` o entrambi. Se si specifica `OpenMonitoring`, non è possibile disabilitare i due esportatori contemporaneamente.

Abilita il monitoraggio aperto sul cluster MSK Provisioned esistente

Per abilitare il monitoraggio aperto, assicuratevi che il cluster MSK Provisioned sia nello stato. `ACTIVE`

Utilizzando il AWS Management Console

1. Accedere a e aprire la console Amazon MSK da https://console.aws.amazon.com/msk/casa?AWS Management Console region=us-east-1#/home/.
2. Scegliere il nome del cluster da aggiornare. In questo modo si accede alla pagina dei dettagli del cluster.
3. Nella scheda Proprietà, scorri verso il basso per trovare la sezione Monitoraggio.
4. Scegli Modifica.

5. Selezionare la casella di controllo accanto a Enable open monitoring with Prometheus (Abilita monitoraggio aperto con Prometheus).
6. Scegli Save changes (Salva modifiche).

Usando il AWS CLI

- Richiama il comando update-monitoring e specifica l'opzione `open-monitoring`. Abilitare `JmxExporter`, `NodeExporter` o entrambi. Se si specifica `open-monitoring`, non è possibile disabilitare i due esportatori contemporaneamente.

Utilizzo dell'API

- Richiama l'UpdateMonitoringoperazione e specifica`OpenMonitoring`. Abilitare `jmxExporter`, `nodeExporter` o entrambi. Se si specifica `OpenMonitoring`, non è possibile disabilitare i due esportatori contemporaneamente.

Configurare un host Prometheus su un'istanza Amazon EC2

Questa procedura descrive come configurare un host Prometheus utilizzando un file prometheus.yml.

1. Scarica il server Prometheus dalla https://prometheus.io/download/#prometheus tua istanza Amazon. EC2
2. Estrarre il file scaricato in una directory e passare a tale directory.
3. Creare un file denominato `prometheus.yml` con i seguenti contenuti:

```
# file: prometheus.yml
# my global config
global:
  scrape_interval:     60s

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped
 from this config.
  - job_name: 'prometheus'
    static_configs:
    # 9090 is the prometheus server port
    - targets: ['localhost:9090']
```

```
- job_name: 'broker'
  file_sd_configs:
  - files:
    - 'targets.json'
```

4. Usa l'ListNodesoperazione per ottenere un elenco dei broker del tuo cluster.
5. Creare un file denominato `targets.json` con il seguente JSON. Sostituisci *broker_dns_1broker_dns_2*, e il resto dei nomi DNS dei broker con i nomi DNS che hai ottenuto per i tuoi broker nel passaggio precedente. Includi tutti i broker ottenuti nel passaggio precedente. Amazon MSK utilizza la porta 11001 per JMX Exporter e la porta 11002 per Node Exporter.

   ZooKeeper mode targets.json

```
[
  {
    "labels": {
      "job": "jmx"
    },
    "targets": [
      "broker_dns_1:11001",
      "broker_dns_2:11001",
      .
      .
      .
      "broker_dns_N:11001"
    ]
  },
  {
    "labels": {
      "job": "node"
    },
    "targets": [
      "broker_dns_1:11002",
      "broker_dns_2:11002",
      .
      .
      .
      "broker_dns_N:11002"
    ]
  }
]
```

KRaft mode targets.json

```
[
  {
    "labels": {
      "job": "jmx"
    },
    "targets": [
      "broker_dns_1:11001",
      "broker_dns_2:11001",
      .
      .
      .
      "broker_dns_N:11001",
      "controller_dns_1:11001",
      "controller_dns_2:11001",
      "controller_dns_3:11001"
    ]
  },
  {
    "labels": {
      "job": "node"
    },
    "targets": [
      "broker_dns_1:11002",
      "broker_dns_2:11002",
      .
      .
      .
      "broker_dns_N:11002"
    ]
  }
]
```

> **Note**
>
> Per estrarre le metriche JMX dai KRaft controller, aggiungi i nomi DNS dei controller come destinazioni nel file JSON. Ad esempio: sostituendo `controller_dns_1` con il nome `controller_dns_1:11001` DNS effettivo del controller.

6. Per avviare il server Prometheus sulla tua istanza EC2 Amazon, esegui il seguente comando nella directory in cui hai estratto i file Prometheus e salvato e. `prometheus.yml` `targets.json`

```
./prometheus
```

7. Trova l'indirizzo IP IPv4 pubblico dell' EC2 istanza Amazon su cui hai eseguito Prometheus nel passaggio precedente. Questo indirizzo IP pubblico è necessario nella fase seguente.
8. Per accedere all'interfaccia utente web di Prometheus, apri un browser in grado di accedere alla tua istanza EC2 Amazon e vai *`Prometheus-Instance-Public-IP`*`:9090` a, *`Prometheus-Instance-Public-IP`* dov'è l'indirizzo IP pubblico che hai ottenuto nel passaggio precedente.

Usa le metriche di Prometheus

Tutti i parametri inviati da Apache Kafka a JMX sono accessibili tramite il monitoraggio aperto con Prometheus. Per informazioni sui parametri Apache Kafka, consulta la sezione relativa al monitoraggio nella documentazione di Apache Kafka. Oltre alle metriche di Apache Kafka, le metriche del consumer-lag sono disponibili anche sulla porta 11001 con il nome JMX. MBean `kafka.consumer.group:type=ConsumerLagMetrics` Puoi anche utilizzare Prometheus Node Exporter per ottenere i parametri della CPU e del disco per i tuoi broker sulla porta 11002.

Archivia i parametri di Prometheus in Amazon Managed Service for Prometheus

Servizio gestito da Amazon per Prometheus è un servizio di monitoraggio e avviso compatibile con Prometheus che puoi utilizzare per monitorare i cluster Amazon MSK. È un servizio completamente gestito che dimensiona automaticamente l'importazione, l'archiviazione, le query e gli avvisi dei parametri. Si integra inoltre con i servizi AWS di sicurezza per offrirti un accesso rapido e sicuro ai tuoi dati. È possibile utilizzare il linguaggio di query open source ProMQL per fare una query e creare avvisi relativi ai parametri.

Per ulteriori informazioni, consultare Guida introduttiva ad Amazon Managed Service for Prometheus.

Monitora i ritardi dei consumatori

Il monitoraggio del ritardo dei consumatori consente di identificare i consumatori lenti o bloccati che non tengono il passo con i dati più recenti disponibili su un argomento. Se necessario, puoi quindi intraprendere operazioni correttive, come il dimensionamento o il riavvio di tali consumatori. Per monitorare il ritardo dei consumatori, puoi utilizzare Amazon CloudWatch o open monitoring with Prometheus.

I parametri relativi al ritardo dei consumatori quantificano la differenza tra i dati più recenti scritti sui tuoi argomenti e i dati letti dalle tue applicazioni. Amazon MSK fornisce le seguenti metriche relative al ritardo dei consumatori, che puoi ottenere tramite Amazon CloudWatch o tramite il monitoraggio aperto con Prometheus:,,, e. `EstimatedMaxTimeLag` `EstimatedTimeLag` `MaxOffsetLag` `OffsetLag` `SumOffsetLag` Per ulteriori informazioni su questi parametri, consulta the section called “CloudWatch metriche per i broker Standard”.

Amazon MSK supporta i parametri di ritardo dei consumatori per i cluster con Apache Kafka 2.2.1 o una versione successiva. Quando lavori con Kafka e le metriche, considera i seguenti punti: CloudWatch

- Le metriche relative al ritardo dei consumatori vengono emesse solo se un gruppo di consumatori si trova in uno stato STABILE o VUOTO. Un gruppo di consumatori è STABILE dopo il completamento con successo del riequilibrio, garantendo che le partizioni siano distribuite uniformemente tra i consumatori.
- Le metriche relative al ritardo dei consumatori sono assenti nei seguenti scenari:
  - Se il gruppo di consumatori è instabile.
  - Il nome del gruppo di consumatori contiene i due punti (:).
  - Non hai impostato l'offset di consumo per il gruppo di consumatori.
- I nomi dei gruppi di consumatori vengono utilizzati come dimensioni per le metriche relative al ritardo dei consumatori in. CloudWatch Sebbene Kafka supporti i caratteri UTF-8 nei nomi dei gruppi di consumatori, CloudWatch supporta solo caratteri ASCII per i valori delle dimensioni. Se si utilizzano caratteri non ASCII nei nomi dei gruppi di consumatori, elimina le metriche relative al ritardo dei consumatori. CloudWatch Per assicurarti che le metriche relative al ritardo dei consumatori vengano acquisite correttamente CloudWatch, devi utilizzare solo caratteri ASCII nei nomi dei gruppi di consumatori.

Usa gli avvisi sulla capacità di archiviazione di Amazon MSK

Nei cluster con provisioning di Amazon MSK, scegli la capacità di archiviazione principale del cluster. L'esaurimento della capacità di archiviazione di un broker nel cluster con provisioning può influire sulla sua capacità di produrre e consumare dati, causando costosi tempi di inattività. Amazon MSK offre CloudWatch parametri per aiutarti a monitorare la capacità di storage del cluster. Inoltre, Amazon MSK invia automaticamente avvisi dinamici sulla capacità di archiviazione del cluster in modo da semplificare il rilevamento e la risoluzione dei problemi correlati. Gli avvisi sulla capacità di archiviazione includono raccomandazioni sulle misure a breve e a lungo termine necessarie

per gestire la capacità di archiviazione del cluster. Dalla console Amazon MSK, puoi utilizzare i collegamenti rapidi all'interno degli avvisi per intraprendere immediatamente le operazioni consigliate.

Esistono due tipi di avvisi MSK sulla capacità di archiviazione: proattivi e correttivi.

- Gli avvisi proattivi sulla capacità di archiviazione ("Operazione richiesta") segnalano i potenziali problemi di archiviazione del cluster. Quando un broker in un cluster MSK ha utilizzato oltre il 60% o l'80% della sua capacità di archiviazione su disco, riceverai avvisi proattivi per il broker interessato.
- Gli avvisi correttivi relativi alla capacità di archiviazione ("Operazione critica richiesta") prevedono l'adozione di misure correttive per risolvere un problema critico del cluster quando uno dei broker del cluster MSK ha esaurito la capacità di archiviazione su disco.

Amazon MSK invia automaticamente questi avvisi alla console Amazon MSK,AWS Health Dashboard EventBridge, Amazon e ai contatti e-mail del tuo account. AWS Puoi anche configurare Amazon EventBridge per inviare questi avvisi a Slack o a strumenti come New Relic e Datadog.

Gli avvisi sulla capacità di archiviazione sono abilitati per impostazione predefinita per tutti i cluster MSK con provisioning e non possono essere disattivati. Questa funzionalità è supportata in tutte le regioni in cui è disponibile MSK.

Monitora gli avvisi sulla capacità di archiviazione

Puoi verificare la presenza di avvisi sulla capacità di archiviazione in diversi modi:

- Accedi alla console Amazon MSK. Gli avvisi sulla capacità di archiviazione vengono visualizzati nel riquadro degli avvisi del cluster per 90 giorni. Gli avvisi contengono raccomandazioni e operazioni da eseguire con un solo clic per risolvere i problemi di capacità di archiviazione su disco.
- Usa ListClusters, ListClustersV2 o DescribeClusterV2 APIs per visualizzare `CustomerActionStatus` tutti gli avvisi relativi a un cluster. DescribeCluster
- Vai ad AWS Health Dashboard per visualizzare gli avvisi di MSK e di altri servizi AWS .
- Configura AWS Health API e Amazon EventBridge per indirizzare le notifiche di avviso a piattaforme di terze parti come Datadog e NewRelic Slack.

Aggiornamento delle impostazioni di sicurezza di un cluster Amazon MSK

Utilizza l'operazione UpdateSecurityAmazon MSK per aggiornare le impostazioni di autenticazione e crittografia client-broker del tuo cluster MSK. Puoi anche aggiornare la Private Security

Authority utilizzata per firmare i certificati per l'autenticazione TLS reciproca. Non puoi modificare l'impostazione di crittografia in-cluster (). broker-to-broker

Per poter aggiornare le impostazioni di sicurezza, il cluster deve essere nello stato `ACTIVE`.

Se attivi l'autenticazione tramite IAM, SASL o TLS, devi attivare anche la crittografia tra client e broker. La tabella di seguito riporta le possibili combinazioni.

Autenticazione	Opzioni di crittografia client-broker	Crittografia broker-broker
Unauthenticated	TLS, PLAINTEXT, TLS_PLAINTEXT	Può essere attiva o non attiva.
mTLS	TLS, TLS_PLAINTEXT	Deve essere attiva.
SASL/SCRAM	TLS	Deve essere attiva.
SASL/IAM	TLS	Deve essere attiva.

Quando la crittografia client-broker è impostata su `TLS_PLAINTEXT` e l'autenticazione client è impostata su `mTLS`, Amazon MSK crea due tipi di ascoltatore a cui i client possono connettersi: un ascoltatore a cui i client possono connettersi utilizzando l'autenticazione mTLS con crittografia TLS e un altro a cui i client possono connettersi senza autenticazione o crittografia (non crittografato).

Per ulteriori informazioni sulle impostazioni di sicurezza, consulta la sezione the section called “Sicurezza”.

Aggiorna le impostazioni di sicurezza del cluster Amazon MSK utilizzando AWS Management Console

1. Accedere a e aprire la console Amazon MSK da https://console.aws.amazon.com/msk/casa? AWS Management Console region=us-east-1#/home/.
2. Seleziona il cluster MSK che desideri aggiornare.
3. Nella sezione Impostazioni di sicurezza, scegli Modifica.
4. Scegli le impostazioni di autenticazione e crittografia che desideri per il cluster, quindi seleziona Salva modifiche.

Aggiornamento delle impostazioni di sicurezza del cluster Amazon MSK utilizzando AWS CLI

1. Crea un file JSON contenente le impostazioni di crittografia che desideri assegnare al cluster. Di seguito è riportato un esempio.

   > **Note**
   >
   > È possibile aggiornare solo l'impostazione di crittografia client-broker. Non puoi aggiornare l'impostazione di crittografia in-cluster (broker-to-broker).

   ```
   {"EncryptionInTransit":{"ClientBroker": "TLS"}}
   ```

2. Crea un file JSON contenente le impostazioni di autenticazione che desideri che il cluster utilizzi. Di seguito è riportato un esempio.

   ```
   {"Sasl":{"Scram":{"Enabled":true}}}
   ```

3. Esegui il AWS CLI comando seguente:

   ```
   aws kafka update-security --cluster-arn ClusterArn --current-version Current-Cluster-Version --client-authentication file://Path-to-Authentication-Settings-JSON-File --encryption-info file://Path-to-Encryption-Settings-JSON-File
   ```

   L'output di questa operazione `update-security` è simile al seguente JSON.

   ```
   {

       "ClusterArn": "arn:aws:kafka:us-east-1:012345678012:cluster/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2",
       "ClusterOperationArn": "arn:aws:kafka:us-east-1:012345678012:cluster-operation/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2/0123abcd-abcd-4f7f-1234-9876543210ef"
   }
   ```

4. Per visualizzare lo stato dell'`update-security`operazione, esegui il comando seguente, sostituendolo *`ClusterOperationArn`* con l'ARN ottenuto nell'output del `update-security` comando.

```
aws kafka describe-cluster-operation --cluster-operation-arn ClusterOperationArn
```

L'output di questo comando `describe-cluster-operation` è simile all'esempio JSON seguente.

```
{
    "ClusterOperationInfo": {
        "ClientRequestId": "c0b7af47-8591-45b5-9c0c-909a1a2c99ea",
        "ClusterArn": "arn:aws:kafka:us-east-1:012345678012:cluster/
exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2",
        "CreationTime": "2021-09-17T02:35:47.753000+00:00",
        "OperationArn": "arn:aws:kafka:us-east-1:012345678012:cluster-
operation/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2/0123abcd-
abcd-4f7f-1234-9876543210ef",
        "OperationState": "PENDING",
        "OperationType": "UPDATE_SECURITY",
        "SourceClusterInfo": {},
        "TargetClusterInfo": {}
    }
}
```

Se il valore di `OperationState` è `PENDING` oppure `UPDATE_IN_PROGRESS`, attendi qualche minuto, quindi esegui nuovamente il comando `describe-cluster-operation`.

> **Note**
>
> Le operazioni AWS CLI e le API per l'aggiornamento delle impostazioni di sicurezza di un cluster sono idempotenti. Ciò significa che se si richiama l'operazione di aggiornamento della sicurezza e si specifica un'impostazione di autenticazione o crittografia uguale a quella attualmente utilizzata dal cluster, tale impostazione non verrà modificata.

Aggiornamento delle impostazioni di sicurezza di un cluster tramite l'API

Per aggiornare le impostazioni di sicurezza per un cluster Amazon MSK utilizzando l'API, consulta UpdateSecurity.

> **Note**
>
> Le operazioni AWS CLI e le API per l'aggiornamento delle impostazioni di sicurezza di un cluster MSK sono idempotenti. Ciò significa che se si richiama l'operazione di aggiornamento della sicurezza e si specifica un'impostazione di autenticazione o crittografia uguale a quella attualmente utilizzata dal cluster, tale impostazione non verrà modificata.

Espandi il numero di broker in un cluster Amazon MSK

Utilizza questa operazione di Amazon MSK quando desideri incrementare il numero di broker nel cluster MSK. Per espandere un cluster, assicurati che il suo stato sia `ACTIVE`.

> **Important**
>
> Se desideri espandere un cluster MSK, assicurati di utilizzare questa operazione di Amazon MSK. Non provare ad aggiungere broker a un cluster senza utilizzare questa operazione.

Per informazioni su come ribilanciare le partizioni dopo aver aggiunto broker a un cluster, consulta the section called “Riassegnazione delle partizioni”.

Espandi un cluster Amazon MSK utilizzando AWS Management Console

Questo processo descrive come aumentare il numero di broker in un cluster Amazon MSK utilizzando il. AWS Management Console

1. Accedere a e aprire la console Amazon MSK da https://console.aws.amazon.com/msk/casa? AWS Management Console region=us-east-1#/home/.
2. Scegli il cluster MSK di cui desideri aumentare numero di broker.
3. Dal menu a discesa Azioni, scegli Modifica numero di broker.
4. Inserisci il numero di broker di cui deve disporre il cluster per zona di disponibilità, quindi scegli Salva modifiche.

Espandi un cluster Amazon MSK utilizzando AWS CLI

Questo processo descrive come aumentare il numero di broker in un cluster Amazon MSK utilizzando il. AWS CLI

1. Esegui il comando seguente, sostituendolo *ClusterArn* con l'Amazon Resource Name (ARN) che hai ottenuto quando hai creato il cluster. Se non disponi dell'ARN per il cluster, puoi trovarlo elencando tutti i cluster. Per ulteriori informazioni, consulta the section called “Elenca i cluster”.

   Sostituiscilo *Current-Cluster-Version* con la versione corrente del cluster.

   > **Important**
   >
   > Le versioni del cluster non sono interi semplici. Per trovare la versione corrente del cluster, usa l'DescribeClusteroperazione o il comando AWS CLI describe-cluster. Una versione di esempio è `KTVPDKIKX0DER`.

   Il *Target-Number-of-Brokers* parametro rappresenta il numero totale di nodi broker che si desidera che il cluster disponga quando l'operazione viene completata correttamente. Il valore specificato *Target-Number-of-Brokers* deve essere un numero intero maggiore del numero corrente di broker nel cluster. Deve anche essere un multiplo del numero di zone di disponibilità.

   ```
   aws kafka update-broker-count --cluster-arn ClusterArn --current-version Current-Cluster-Version --target-number-of-broker-nodes Target-Number-of-Brokers
   ```

   L'output di questa operazione `update-broker-count` è simile al seguente JSON.

   ```
   {

       "ClusterArn": "arn:aws:kafka:us-east-1:012345678012:cluster/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2",
       "ClusterOperationArn": "arn:aws:kafka:us-east-1:012345678012:cluster-operation/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2/0123abcd-abcd-4f7f-1234-9876543210ef"
   }
   ```

2. Per ottenere il risultato dell'`update-broker-count`operazione, esegui il comando seguente, sostituendolo *ClusterOperationArn* con l'ARN ottenuto nell'output del `update-broker-count` comando.

   ```
   aws kafka describe-cluster-operation --cluster-operation-arn ClusterOperationArn
   ```

L'output di questo comando `describe-cluster-operation` è simile all'esempio JSON seguente.

```
{
    "ClusterOperationInfo": {
        "ClientRequestId": "c0b7af47-8591-45b5-9c0c-909a1a2c99ea",
        "ClusterArn": "arn:aws:kafka:us-east-1:012345678012:cluster/
exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2",
        "CreationTime": "2019-09-25T23:48:04.794Z",
        "OperationArn": "arn:aws:kafka:us-east-1:012345678012:cluster-
operation/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2/0123abcd-
abcd-4f7f-1234-9876543210ef",
        "OperationState": "UPDATE_COMPLETE",
        "OperationType": "INCREASE_BROKER_COUNT",
        "SourceClusterInfo": {
            "NumberOfBrokerNodes": 9
        },
        "TargetClusterInfo": {
            "NumberOfBrokerNodes": 12
        }
    }
}
```

In questo output, `OperationType` è `INCREASE_BROKER_COUNT`. Se il valore di `OperationState` è `UPDATE_IN_PROGRESS`, attendi qualche minuto, quindi esegui nuovamente il comando `describe-cluster-operation`.

Espandi un cluster Amazon MSK utilizzando l'API

Per aumentare il numero di broker in un cluster utilizzando l'API, consulta. UpdateBrokerCount

Rimuovere un broker da un cluster Amazon MSK

Usa questa operazione Amazon MSK quando desideri rimuovere broker dai cluster con provisioning di Amazon Managed Streaming for Apache Kafka (MSK). Puoi ridurre la capacità di storage e di elaborazione del cluster rimuovendo set di broker, senza alcun impatto sulla disponibilità, rischio di durabilità dei dati o interruzione delle applicazioni di streaming di dati.

Puoi aggiungere altri broker al cluster per gestire l'aumento del traffico e rimuovere i broker quando il traffico diminuisce. Grazie alla funzionalità di aggiunta e rimozione dei broker, è possibile utilizzare al

meglio la capacità del cluster e ottimizzare i costi dell'infrastruttura MSK. La rimozione dei broker offre il controllo a livello di broker sulla capacità del cluster esistente per soddisfare le esigenze di carico di lavoro ed evitare la migrazione verso un altro cluster.

Utilizza la AWS console, l'interfaccia a riga di comando (CLI), l'SDK o AWS CloudFormation per ridurre il numero di broker del cluster fornito. MSK seleziona i broker che non dispongono di alcuna partizione (ad eccezione degli argomenti Canary) e impedisce alle applicazioni di produrre dati per tali broker, rimuovendo al contempo in modo sicuro tali broker dal cluster.

È necessario rimuovere un broker per zona di disponibilità, se si desidera ridurre lo storage e l'elaborazione di un cluster. Ad esempio, è possibile rimuovere due broker da un cluster con due zone di disponibilità o tre broker da un cluster con tre zone di disponibilità in un'unica operazione di rimozione dei broker.

Per informazioni su come ribilanciare le partizioni dopo aver rimosso i broker da un cluster, vedere. the section called “Riassegnazione delle partizioni”

È possibile rimuovere i broker da tutti i cluster con provisioning MSK basati su M5 e M7g, indipendentemente dalla dimensione dell'istanza.

La rimozione dei broker è supportata nelle versioni di Kafka 2.8.1 e successive, inclusi i cluster modali. KRaft

Argomenti

Preparati a rimuovere i broker rimuovendo tutte le partizioni

Prima di iniziare il processo di rimozione del broker, sposta innanzitutto tutte le partizioni, tranne quelle relative agli argomenti `__amazon_msk_canary` e ai broker che `__amazon_msk_canary_state` intendi rimuovere. Si tratta di argomenti interni creati da Amazon MSK per i parametri diagnostici e di salute dei cluster.

Puoi utilizzare Kafka admin APIs o Cruise Control per spostare le partizioni su altri broker che intendi mantenere nel cluster. Vedi Riassegnare le partizioni.

Procedura di esempio per rimuovere le partizioni

Questa sezione è un esempio di come rimuovere le partizioni dal broker che intendi rimuovere. Supponiamo di avere un cluster con 6 broker, 2 broker in ogni AZ e che abbia quattro argomenti:

- `__amazon_msk_canary`
- `__consumer_offsets`
- `__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-c657f7e4ff32-2`
- `msk-brk-rmv`

1. Crea una macchina client come descritto in Creare una macchina client.
2. Dopo aver configurato il computer client, esegui il comando seguente per elencare tutti gli argomenti disponibili nel cluster.

   ```
   ./bin/kafka-topics.sh --bootstrap-server “CLUSTER_BOOTSTRAP_STRING” --list
   ```

   In questo esempio, vediamo quattro nomi di argomenti,`__amazon_msk_canary`, `__consumer_offsets__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-c657f7e4ff32-2`, e`msk-brk-rmv`.
3. Crea un file json chiamato `topics.json` sul computer client e aggiungi tutti i nomi degli argomenti utente come nel seguente esempio di codice. Non è necessario includere il nome dell'`__amazon_msk_canary`argomento in quanto si tratta di un argomento gestito dal servizio che verrà spostato automaticamente quando necessario.

   ```
   {
   "topics": [
   {"topic": "msk-brk-rmv"},
   {"topic": "__consumer_offsets"},
   {"topic": "__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-c657f7e4ff32-2"}
   ],
   "version":1
   }
   ```
4. Esegui il comando seguente per generare una proposta per spostare le partizioni su soli 3 broker su 6 broker del cluster.

```
./bin/kafka-reassign-partitions.sh --bootstrap-server “CLUSTER_BOOTSTRAP_STRING” --
topics-to-move-json-file topics.json --broker-list 1,2,3 --generate
```

5. Crea un file chiamato `reassignment-file.json` e copia il comando `proposed partition reassignment configuration` che hai ottenuto dal precedente comando.
6. Esegui il seguente comando per spostare le partizioni specificate in. `reassignment-file.json`

   ```
   ./bin/kafka-reassign-partitions.sh --bootstrap-server “CLUSTER_BOOTSTRAP_STRING” --
   reassignment-json-file reassignment-file.json --execute
   ```

   L'esito si presenta in maniera analoga all'immagine riportata di seguito.

   ```
   Successfully started partition reassignments for morpheus-test-topic-1-0,test-
   topic-1-0
   ```

7. Esegui il comando seguente per verificare che tutte le partizioni siano state spostate.

   ```
   ./bin/kafka-reassign-partitions.sh --bootstrap-server “CLUSTER_BOOTSTRAP_STRING” --
   reassignment-json-file reassignment-file.json --verify
   ```

   L'output è simile al seguente. Monitora lo stato fino a quando tutte le partizioni negli argomenti richiesti non sono state riassegnate correttamente:

   ```
   Status of partition reassignment:
   Reassignment of partition msk-brk-rmv-0 is completed.
   Reassignment of partition msk-brk-rmv-1 is completed.
   Reassignment of partition __consumer_offsets-0 is completed.
   Reassignment of partition __consumer_offsets-1 is completed.
   ```

8. Quando lo stato indica che la riassegnazione delle partizioni per ogni partizione è stata completata, monitora le `UserPartitionExists` metriche per 5 minuti per assicurarti che vengano visualizzate dai broker da cui hai spostato le `0` partizioni. Dopo averlo confermato, puoi procedere alla rimozione del broker dal cluster.

Rimuovi un broker con la console di AWS gestione

Per rimuovere i broker con la console di gestione AWS

1. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/.
2. Scegli il cluster MSK che contiene i broker che desideri rimuovere.
3. Nella pagina dei dettagli del cluster, scegli il pulsante Azioni e seleziona l'opzione Modifica numero di broker.
4. Inserisci il numero di broker che desideri che il cluster abbia per zona di disponibilità. La console riepiloga il numero di broker nelle zone di disponibilità che verranno rimossi. Assicurati che sia quello che vuoi.
5. Scegli Save changes (Salva modifiche).

Per evitare la rimozione accidentale del broker, la console ti chiede di confermare che desideri eliminare i broker.

Rimuovi un broker con la AWS CLI

Esegui il comando seguente, sostituendolo `ClusterArn` con l'Amazon Resource Name (ARN) che hai ottenuto quando hai creato il cluster. Se non disponi dell'ARN per il cluster, puoi trovarlo elencando tutti i cluster. Per ulteriori informazioni, consulta Listing Amazon MSK clusters. Sostituisci `Current-Cluster-Version` con la versione corrente del cluster.

> **Important**
>
> Le versioni del cluster non sono interi semplici. Per trovare la versione corrente del cluster, usa l'DescribeClusteroperazione o il comando AWS CLI describe-cluster. Una versione di esempio è `KTVPDKIKX0DER`.

Il *`Target-Number-of-Brokers`* parametro rappresenta il numero totale di nodi broker che si desidera che il cluster disponga quando l'operazione viene completata correttamente. Il valore specificato *`Target-Number-of-Brokers`* deve essere un numero intero inferiore al numero corrente di broker nel cluster. Deve anche essere un multiplo del numero di zone di disponibilità.

```
aws kafka update-broker-count --cluster-arn ClusterArn --current-version Current-
Cluster-Version --target-number-of-broker-nodes Target-Number-of-Brokers
```

L'output di questa operazione `update-broker-count` è simile al seguente JSON.

```
{
"ClusterOperationInfo": {
"ClientRequestId": "c0b7af47-8591-45b5-9c0c-909a1a2c99ea",
        "ClusterArn": "arn:aws:kafka:us-east-1:012345678012:cluster/exampleClusterName/
abcdefab-1234-abcd-5678-cdef0123ab01-2",
        "CreationTime": "2019-09-25T23:48:04.794Z",
        "OperationArn": "arn:aws:kafka:us-east-1:012345678012:cluster-
operation/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2/0123abcd-
abcd-4f7f-1234-9876543210ef",
        "OperationState": "UPDATE_COMPLETE",
        "OperationType": "DECREASE_BROKER_COUNT",
        "SourceClusterInfo": {
"NumberOfBrokerNodes": 12
        },
        "TargetClusterInfo": {
"NumberOfBrokerNodes": 9
        }
    }
}
```

In questo output, `OperationType` è `DECREASE_BROKER_COUNT`. Se il valore di `OperationState` è `UPDATE_IN_PROGRESS`, attendi qualche minuto, quindi esegui nuovamente il comando `describe-cluster-operation`.

Rimuovi un broker con l'API AWS

Per rimuovere i broker in un cluster utilizzando l'API, consulta UpdateBrokerCountAmazon Managed Streaming for Apache Kafka API Reference.

Aggiornamento delle dimensioni del broker del cluster Amazon MSK

Puoi scalare il tuo cluster MSK su richiesta modificando le dimensioni dei broker senza riassegnare le partizioni Apache Kafka. La modifica delle dimensioni dei broker offre la flessibilità necessaria per adattare la capacità di calcolo del cluster MSK in base alle variazioni dei carichi di lavoro, senza interrompere l'I/O del cluster. Amazon MSK utilizza le stesse dimensioni di broker per tutti i broker di un determinato cluster.

Per i broker standard, puoi aggiornare le dimensioni del broker del cluster da M5 o T3 a M7g, da T3 a M5 o da M7g a M5.

> **Note**
>
> Non è possibile migrare da un broker di dimensioni maggiori a un broker di dimensioni inferiori. Ad esempio, da M7g.large a T3.small.

Per i broker Express, puoi utilizzare solo le dimensioni dei broker M7g.

Questo argomento descrive come aggiornare le dimensioni del broker per il cluster MSK.

Tieni presente che la migrazione a un broker di dimensioni inferiori può ridurre le prestazioni e ridurre il throughput massimo raggiungibile per broker. La migrazione a un broker di dimensioni maggiori può aumentare le prestazioni ma potrebbe costare di più.

L'aggiornamento delle dimensioni di un broker avviene in modo continuativo mentre il cluster è attivo e funzionante. Ciò significa che Amazon MSK disattiva un broker alla volta per eseguire l'aggiornamento delle dimensioni del broker. Per informazioni su come rendere altamente disponibile un cluster durante un aggiornamento delle dimensioni di un broker, consulta. the section called “Creazione di cluster a disponibilità elevata” Per ridurre ulteriormente il potenziale impatto sulla produttività, è possibile eseguire l'aggiornamento delle dimensioni del broker durante un periodo di traffico ridotto.

Durante un aggiornamento delle dimensioni di un broker, puoi continuare a produrre e consumare dati. Tuttavia, è necessario attendere il completamento dell'aggiornamento prima di poter riavviare i broker o richiamare una delle operazioni di aggiornamento elencate nelle operazioni di Amazon MSK.

Se desideri aggiornare il cluster a un broker di dimensioni inferiori, ti consigliamo di provare prima l'aggiornamento su un cluster di test per vedere come influisce sullo scenario.

> **Important**
>
> Non puoi aggiornare un cluster a un broker di dimensioni inferiori se il numero di partizioni per broker supera il numero massimo specificato in. the section called “ Dimensionamento corretto del cluster: numero di partizioni per broker standard”

Argomenti

- Aggiornamento delle dimensioni del broker tramite l'API

Aggiorna le dimensioni del broker del cluster Amazon MSK utilizzando AWS Management Console

Questo processo mostra come aggiornare le dimensioni del broker del cluster Amazon MSK utilizzando il AWS Management Console

1. Accedere a e aprire la console Amazon MSK da https://console.aws.amazon.com/msk/casa? AWS Management Console region=us-east-1#/home/.
2. Scegliete il cluster MSK per il quale desiderate aggiornare le dimensioni del broker.
3. Nella pagina dei dettagli del cluster, trova la sezione di riepilogo dei broker e scegli Modifica le dimensioni del broker.
4. Scegli la dimensione del broker che desideri dall'elenco.
5. Salva le modifiche.

Aggiorna le dimensioni del broker del cluster Amazon MSK utilizzando AWS CLI

Esegui il comando seguente, sostituendolo *ClusterArn* con l'Amazon Resource Name (ARN) che hai ottenuto quando hai creato il cluster. Se non disponi dell'ARN per il cluster, puoi trovarlo elencando tutti i cluster. Per ulteriori informazioni, consulta the section called “Elenca i cluster”.

1. Sostituiscilo *Current-Cluster-Version* con la versione corrente del cluster e *TargetType* con la nuova dimensione che desideri che abbiano i broker. Per ulteriori informazioni sulle dimensioni dei broker, consultathe section called “Tipi di broker”.

   ```
   aws kafka update-broker-type --cluster-arn ClusterArn --current-version Current-Cluster-Version --target-instance-type TargetType
   ```

   Di seguito è riportato un esempio di come utilizzare questo comando:

   ```
   aws kafka update-broker-type --cluster-arn "arn:aws:kafka:us-east-1:0123456789012:cluster/exampleName/abcd1234-0123-abcd-5678-1234abcd-1" --current-version "K1X5R6FKA87" --target-instance-type kafka.m5.large
   ```

   L'output di questo comando è simile all'esempio JSON seguente.

```
{
    "ClusterArn": "arn:aws:kafka:us-east-1:0123456789012:cluster/exampleName/
abcd1234-0123-abcd-5678-1234abcd-1",
    "ClusterOperationArn": "arn:aws:kafka:us-east-1:012345678012:cluster-
operation/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2/0123abcd-
abcd-4f7f-1234-9876543210ef"
}
```

2. Per ottenere il risultato dell'`update-broker-type`operazione, esegui il comando seguente, sostituendolo *`ClusterOperationArn`* con l'ARN ottenuto nell'output del `update-broker-type` comando.

```
aws kafka describe-cluster-operation --cluster-operation-arn ClusterOperationArn
```

L'output di questo comando `describe-cluster-operation` è simile all'esempio JSON seguente.

```
{
  "ClusterOperationInfo": {
    "ClientRequestId": "982168a3-939f-11e9-8a62-538df00285db",
    "ClusterArn": "arn:aws:kafka:us-east-1:0123456789012:cluster/exampleName/
abcd1234-0123-abcd-5678-1234abcd-1",
    "CreationTime": "2021-01-09T02:24:22.198000+00:00",
    "OperationArn": "arn:aws:kafka:us-east-1:012345678012:cluster-operation/
exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2/0123abcd-
abcd-4f7f-1234-9876543210ef",
    "OperationState": "UPDATE_COMPLETE",
    "OperationType": "UPDATE_BROKER_TYPE",
    "SourceClusterInfo": {
      "InstanceType": "t3.small"
    },
    "TargetClusterInfo": {
      "InstanceType": "m5.large"
    }
  }
}
```

Se il valore di `OperationState` è `UPDATE_IN_PROGRESS`, attendi qualche minuto, quindi esegui nuovamente il comando `describe-cluster-operation`.

Aggiornamento delle dimensioni del broker tramite l'API

Per aggiornare le dimensioni del broker utilizzando l'API, consulta UpdateBrokerType.

Puoi utilizzarla `UpdateBrokerType` per aggiornare le dimensioni del broker del cluster da M5 o T3 a M7g o da M7g a M5.

Usa LinkedIn il Cruise Control per Apache Kafka con Amazon MSK

Puoi utilizzare LinkedIn Cruise Control per ribilanciare il cluster Amazon MSK, rilevare e correggere anomalie e monitorare lo stato e l'integrità del cluster.

Download e compilazione di Cruise Control

1. Crea un' EC2 istanza Amazon nello stesso Amazon VPC del cluster Amazon MSK.
2. Installa Prometheus sull'istanza EC2 Amazon che hai creato nel passaggio precedente. Prendi nota dell'IP privato e della porta. Il numero di porta predefinito è 9090. Per informazioni su come configurare Prometheus per aggregare i parametri per un cluster, consulta la pagina the section called “Monitora con Prometheus”.
3. Scarica Cruise Control sull' EC2 istanza Amazon. (In alternativa, puoi utilizzare un' EC2 istanza Amazon separata per Cruise Control, se preferisci.) Per un cluster con Apache Kafka versione 2.4.*, usa la versione 2.4.* di Cruise Control più recente. Se il tuo cluster ha una versione di Apache Kafka precedente alla 2.4.*, utilizza la versione 2.0.* di Cruise Control più recente.
4. Decomprimi il file Cruise Control, quindi vai alla cartella decompressa.
5. Esegui il comando seguente per installare git.

```
sudo yum -y install git
```

6. Esegui il comando seguente per inizializzare il repository locale. Sostituiscila *Your-Cruise-Control-Folder* con il nome della cartella attuale (la cartella che hai ottenuto quando hai decompresso il download di Cruise Control).

```
git init && git add . && git commit -m "Init local repo." && git tag -a Your-Cruise-Control-Folder -m "Init local version."
```

7. Esegui il comando seguente per creare il codice sorgente.

```
./gradlew jar copyDependantLibs
```

Configurazione ed esecuzione di Cruise Control

1. Apporta le seguenti modifiche al file `config/cruisecontrol.properties`. Sostituisci la stringa bootstrap servers e bootstrap-brokers di esempio con i valori del tuo cluster. Per recuperare queste stringhe per il cluster, puoi consultare i dettagli del cluster nella console. In alternativa, puoi utilizzare le operazioni GetBootstrapBrokerse DescribeClusterAPI o i loro equivalenti CLI.

```
# If using TLS encryption, use 9094; use 9092 if using plaintext
bootstrap.servers=b-1.test-cluster.2skv42.c1.kafka.us-
east-1.amazonaws.com:9094,b-2.test-cluster.2skv42.c1.kafka.us-
east-1.amazonaws.com:9094,b-3.test-cluster.2skv42.c1.kafka.us-
east-1.amazonaws.com:9094

# SSL properties, needed if cluster is using TLS encryption
security.protocol=SSL
ssl.truststore.location=/home/ec2-user/kafka.client.truststore.jks

# Use the Prometheus Metric Sampler
metric.sampler.class=com.linkedin.kafka.cruisecontrol.monitor.sampling.prometheus.Prometheu

# Prometheus Metric Sampler specific configuration
prometheus.server.endpoint=1.2.3.4:9090 # Replace with your Prometheus IP and port

# Change the capacity config file and specify its path; details below
capacity.config.file=config/capacityCores.json
```

   Per i broker express, consigliamo di non utilizzarlo `DiskCapacityGoal` in nessuno degli obiettivi configurati nelle configurazioni degli analizzatori.

2. Modifica il file `config/capacityCores.json` per specificare le dimensioni corrette del disco, i core della CPU e i limiti di ingresso/uscita della rete. Per i broker Express, l'inserimento della `DISK` capacità è necessario solo per configurare Cruise Control. Poiché MSK gestisce tutto lo spazio di archiviazione per i broker Express, è necessario impostare questo valore su un numero estremamente alto, ad esempio. `Integer.MAX_VALUE (2147483647)` Per i broker Standard, è possibile utilizzare l'operazione DescribeClusterAPI (o la CLI describe-cluster) per ottenere la dimensione del disco. Per i core della CPU e i limiti di ingresso/uscita di rete, consulta Amazon EC2 Instance Types.

Standard broker config/capacityCores.json

```
{
  "brokerCapacities": [
    {
      "brokerId": "-1",
      "capacity": {
        "DISK": "10000",
        "CPU": {
          "num.cores": "2"
        },
        "NW_IN": "5000000",
        "NW_OUT": "5000000"
      },
      "doc": "This is the default capacity. Capacity unit used for disk is in
 MB, cpu is in number of cores, network throughput is in KB."
    }
  ]
}
```

Express broker config/capacityCores.json

```
{
  "brokerCapacities":[
    {
      "brokerId": "-1",
      "capacity": {
        "DISK": "2147483647",
        "CPU": {"num.cores": "16"},
        "NW_IN": "1073741824",
        "NW_OUT": "1073741824"
      },
      "doc": "This is the default capacity. Capacity unit used for disk is in
 MB, cpu is in number of cores, network throughput is in KB."
    }
  ]
}
```

3. Facoltativamente, puoi installare l'interfaccia utente di Cruise Control. Per scaricarla, consulta la pagina Setting Up Cruise Control Frontend.

4. Esegui il comando seguente per avviar Cruise Control. Prendi in considerazione l'utilizzo di uno strumento come `screen` o `tmux` per mantenere aperta una sessione di lunga durata.

```
<path-to-your-CRUISE-CONTROL-installation>/bin/kafka-cruise-control-start.sh
 config/cruisecontrol.properties 9091
```

5. Usa il Cruise Control APIs o l'interfaccia utente per assicurarti che Cruise Control disponga dei dati di carico del cluster e che fornisca suggerimenti per il ribilanciamento. Potrebbero trascorrere alcuni minuti prima di ottenere una finestra di parametri valida.

> **Important**
>
> Solo le versioni di Cruise Control 2.5.60 e successive sono compatibili con i broker Express, poiché i broker Express non espongono gli endpoint Zookeeper.

Usa il modello di distribuzione automatizzata di Cruise Control per Amazon MSK

Puoi anche utilizzare questo CloudFormation modello per implementare facilmente Cruise Control e Prometheus per ottenere informazioni più approfondite sulle prestazioni del tuo cluster Amazon MSK e ottimizzare l'utilizzo delle risorse.

Caratteristiche principali:

- Provisioning automatico di un' EC2 istanza Amazon con Cruise Control e Prometheus preconfigurati.
- Support per il cluster con provisioning di Amazon MSK.
- Autenticazione flessibile con PlainText e IAM.
- Nessuna dipendenza da Zookeeper per Cruise Control.
- Personalizza facilmente gli obiettivi Prometheus, le impostazioni della capacità del Cruise Control e altre configurazioni fornendo i tuoi file di configurazione memorizzati in un bucket Amazon S3.

Linee guida per il ribilanciamento delle partizioni

Linee guida per la riassegnazione delle partizioni Kafka

La riassegnazione delle partizioni in Kafka può richiedere molte risorse, in quanto comporta il trasferimento di dati significativi tra broker, causando potenzialmente la congestione della rete

e influendo sulle operazioni dei client. Le seguenti best practice consentono di gestire in modo efficace la riassegnazione delle partizioni ottimizzando i tassi di accelerazione, sfruttando i controlli di concorrenza e comprendendo i tipi di riassegnazione per ridurre al minimo le interruzioni delle operazioni del cluster.

Gestione della concorrenza in Cruise Control

Il Cruise Control fornisce parametri di regolazione automatica per controllare la concomitanza dei movimenti di partizione e di leadership. I seguenti parametri aiutano a mantenere un carico accettabile durante le riassegnazioni:

- Movimenti di partizione simultanei massimi: definisci il limite massimo **`num.concurrent.partition.movements.per.broker`** per i movimenti simultanei delle partizioni tra broker, evitando un utilizzo eccessivo della rete.

  Example Esempio

  ```
  num.concurrent.partition.movements.per.broker = 5
  ```

  Questa impostazione limita ogni broker a spostare non più di 10 partizioni alla volta, bilanciando il carico tra i broker.

Usa la limitazione per controllare la larghezza di banda

- Parametro Throttle: quando si esegue la riassegnazione delle partizioni con`kafka-reassign-partitions.sh`, utilizzare `--throttle parameter` per impostare una velocità di trasferimento massima (in byte al secondo) per lo spostamento dei dati tra i broker.

  Example Esempio

  ```
  --throttle 5000000
  ```

  Questo imposta una larghezza di banda massima di 5 MB/s.
- Balance Throttle Settings: La scelta di una frequenza di accelerazione appropriata è fondamentale:

  Se impostato su un valore troppo basso, la riassegnazione potrebbe richiedere molto più tempo.

  Se impostato su un valore troppo alto, i client potrebbero riscontrare un aumento della latenza.

- Inizia con una frequenza di accelerazione conservativa e regola in base al monitoraggio delle prestazioni del cluster. Testa l'acceleratore che hai scelto prima di passare a un ambiente di produzione per trovare l'equilibrio ottimale.

Esegui test e convalida in un ambiente di staging

Prima di implementare le riassegnazioni in produzione, esegui test di carico in un ambiente di staging con configurazioni simili. Ciò consente di ottimizzare i parametri e ridurre al minimo gli impatti imprevisti nella produzione live.

Aggiornamento della configurazione di un cluster Amazon MSK

Per aggiornare la configurazione di un cluster, assicurati che lo stato del cluster sia `ACTIVE`. Inoltre, devi assicurarti che il numero di partizioni per broker sul cluster MSK sia inferiore ai limiti descritti nella sezione the section called " Dimensionamento corretto del cluster: numero di partizioni per broker standard". Non è possibile aggiornare la configurazione di un cluster che supera questi limiti.

Per informazioni sulla configurazione MSK, incluso come creare una configurazione personalizzata, quali proprietà è possibile aggiornare e cosa accade quando si aggiorna la configurazione di un cluster esistente, consulta the section called "Configurazione del broker".

Argomenti

- Disponibilità del broker durante gli aggiornamenti della configurazione
- Aggiornamento della configurazione di un cluster utilizzando AWS CLI
- Aggiorna la configurazione di un cluster Amazon MSK utilizzando l'API

Disponibilità del broker durante gli aggiornamenti della configurazione

Amazon MSK mantiene un'elevata disponibilità durante la maggior parte degli aggiornamenti della configurazione del cluster. Amazon MSK esegue un aggiornamento progressivo in cui aggiorna un broker alla volta. Durante questo processo, il cluster rimane disponibile, anche se i singoli broker verranno riavviati man mano che le relative configurazioni vengono aggiornate. Tuttavia, alcune modifiche alla configurazione potrebbero richiedere l'aggiornamento simultaneo di tutti i broker, il che può causare una breve interruzione a livello di cluster. Per ulteriori informazioni sull'impatto della disponibilità dei broker durante gli aggiornamenti, consulta. Configurazione Amazon MSK Provisioned

Prima di aggiornare i cluster di produzione, ti consigliamo di testare le modifiche alla configurazione in un ambiente non di produzione e di pianificare gli aggiornamenti durante le finestre di manutenzione.

In caso di problemi durante l'aggiornamento del cluster MSK, vedi Come posso risolvere i problemi quando aggiorno il mio cluster Amazon MSK?

Aggiornamento della configurazione di un cluster utilizzando AWS CLI

1. Copiare il JSON seguente e salvarlo in un file. Assegnare un nome al file `configuration-info.json`. Sostituisci *`ConfigurationArn`* con l'Amazon Resource Name (ARN) della configurazione che desideri utilizzare per aggiornare il cluster. La stringa ARN deve essere racchiusa tra virgolette nel seguente JSON.

   Sostituisci *`Configuration-Revision`* con la revisione della configurazione che desideri utilizzare. Le revisioni di configurazione sono interi (numeri interi) che iniziano da 1. Questo intero non deve essere racchiuso tra virgolette nel seguente JSON.

   ```
   {
        "Arn": ConfigurationArn,
        "Revision": Configuration-Revision
   }
   ```

2. Esegui il comando seguente, sostituendolo *`ClusterArn`* con l'ARN ottenuto quando hai creato il cluster. Se non disponi dell'ARN per il cluster, puoi trovarlo elencando tutti i cluster. Per ulteriori informazioni, consulta the section called “Elenca i cluster”.

   Sostituisci *`Path-to-Config-Info-File`* con il percorso del file di informazioni di configurazione. Se hai dato un nome al file creato nel passaggio precedente `configuration-info.json` e lo hai salvato nella directory corrente, allora *`Path-to-Config-Info-File`* è`configuration-info.json`.

   Sostituiscilo *`Current-Cluster-Version`* con la versione corrente del cluster.

   > **Important**
   >
   > Le versioni del cluster non sono interi semplici. Per trovare la versione corrente del cluster, usa l'DescribeClusteroperazione o il comando AWS CLI describe-cluster. Una versione di esempio è `KTVPDKIKX0DER`.

```
aws kafka update-cluster-configuration --cluster-arn ClusterArn --configuration-
info file://Path-to-Config-Info-File --current-version Current-Cluster-Version
```

Di seguito è riportato un esempio di come utilizzare questo comando:

```
aws kafka update-cluster-configuration --cluster-arn "arn:aws:kafka:us-
east-1:0123456789012:cluster/exampleName/abcd1234-0123-abcd-5678-1234abcd-1" --
configuration-info file://c:\users\tester\msk\configuration-info.json --current-
version "K1X5R6FKA87"
```

L'output di questo comando `update-cluster-configuration` è simile all'esempio JSON seguente.

```
{
    "ClusterArn": "arn:aws:kafka:us-east-1:012345678012:cluster/exampleClusterName/
abcdefab-1234-abcd-5678-cdef0123ab01-2",
    "ClusterOperationArn": "arn:aws:kafka:us-east-1:012345678012:cluster-
operation/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2/0123abcd-
abcd-4f7f-1234-9876543210ef"
}
```

3. Per ottenere il risultato dell'`update-cluster-configuration`operazione, esegui il comando seguente, sostituendolo *`ClusterOperationArn`* con l'ARN ottenuto nell'output del `update-cluster-configuration` comando.

```
aws kafka describe-cluster-operation --cluster-operation-arn ClusterOperationArn
```

L'output di questo comando `describe-cluster-operation` è simile all'esempio JSON seguente.

```
{
    "ClusterOperationInfo": {
        "ClientRequestId": "982168a3-939f-11e9-8a62-538df00285db",
        "ClusterArn": "arn:aws:kafka:us-east-1:012345678012:cluster/
exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2",
        "CreationTime": "2019-06-20T21:08:57.735Z",
```

```
        "OperationArn": "arn:aws:kafka:us-east-1:012345678012:cluster-
operation/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2/0123abcd-
abcd-4f7f-1234-9876543210ef",
        "OperationState": "UPDATE_COMPLETE",
        "OperationType": "UPDATE_CLUSTER_CONFIGURATION",
        "SourceClusterInfo": {},
        "TargetClusterInfo": {
            "ConfigurationInfo": {
                "Arn": "arn:aws:kafka:us-east-1:123456789012:configuration/
ExampleConfigurationName/abcdabcd-abcd-1234-abcd-abcd123e8e8e-1",
                "Revision": 1
            }
        }
    }
}
```

In questo output, `OperationType` è `UPDATE_CLUSTER_CONFIGURATION`. Se il valore di `OperationState` è `UPDATE_IN_PROGRESS`, attendi qualche minuto, quindi esegui nuovamente il comando `describe-cluster-operation`.

Aggiorna la configurazione di un cluster Amazon MSK utilizzando l'API

Per utilizzare l'API per aggiornare la configurazione di un cluster Amazon MSK, consulta UpdateClusterConfiguration.

Riavviare un broker per un cluster Amazon MSK

Utilizza questa operazione di Amazon MSK quando desideri riavviare un broker per un cluster MSK. Per riavviare un broker per un cluster, assicurati che il cluster si trovi nello stato `ACTIVE`.

Il servizio Amazon MSK può riavviare i broker del cluster MSK durante la manutenzione del sistema, ad esempio durante l'applicazione di patch o gli aggiornamenti di versione. Il riavvio manuale di un broker consente di testare la resilienza dei client Kafka per determinare come rispondono alla manutenzione del sistema.

Riavviare un broker per un cluster Amazon MSK utilizzando il AWS Management Console

Questo processo descrive come riavviare un broker per un cluster Amazon MSK utilizzando. AWS Management Console

1. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/.
2. Scegli il cluster MSK di cui desideri riavviare il broker.
3. Scorri verso il basso fino alla sezione Dettagli del broker e scegli il broker che desideri riavviare.
4. Scegli il pulsante Riavvia broker.

Riavviare un broker per un cluster Amazon MSK utilizzando il AWS CLI

Questo processo descrive come riavviare un broker per un cluster Amazon MSK utilizzando. AWS CLI

1. Esegui il comando seguente, sostituendolo *`ClusterArn`* con l'Amazon Resource Name (ARN) che hai ottenuto quando hai creato il cluster e *`BrokerId`* poi con l'ID del broker che desideri riavviare.

   > **Note**
   >
   > L'operazione `reboot-broker` supporta il riavvio di un singolo broker alla volta.

   Se non disponi dell'ARN per il cluster, puoi trovarlo elencando tutti i cluster. Per ulteriori informazioni, consulta the section called “Elenca i cluster”.

   Se non disponi del broker IDs per il tuo cluster, puoi trovarlo elencando i nodi del broker. Per ulteriori informazioni, consulta la sezione list-nodes.

   ```
   aws kafka reboot-broker --cluster-arn ClusterArn --broker-ids BrokerId
   ```

   L'output di questa operazione `reboot-broker` è simile al seguente JSON.

   ```
   {

       "ClusterArn": "arn:aws:kafka:us-east-1:012345678012:cluster/exampleClusterName/
   abcdefab-1234-abcd-5678-cdef0123ab01-2",
       "ClusterOperationArn": "arn:aws:kafka:us-east-1:012345678012:cluster-
   operation/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2/0123abcd-
   abcd-4f7f-1234-9876543210ef"
   }
   ```

2. Per ottenere il risultato dell'`reboot-broker`operazione, esegui il comando seguente, sostituendolo *`ClusterOperationArn`* con l'ARN ottenuto nell'output del `reboot-broker` comando.

```
aws kafka describe-cluster-operation --cluster-operation-arn ClusterOperationArn
```

L'output di questo comando `describe-cluster-operation` è simile all'esempio JSON seguente.

```
{
    "ClusterOperationInfo": {
        "ClientRequestId": "c0b7af47-8591-45b5-9c0c-909a1a2c99ea",
        "ClusterArn": "arn:aws:kafka:us-east-1:012345678012:cluster/
exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2",
        "CreationTime": "2019-09-25T23:48:04.794Z",
        "OperationArn": "arn:aws:kafka:us-east-1:012345678012:cluster-
operation/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2/0123abcd-
abcd-4f7f-1234-9876543210ef",
        "OperationState": "REBOOT_IN_PROGRESS",
        "OperationType": "REBOOT_NODE",
        "SourceClusterInfo": {},
        "TargetClusterInfo": {}
    }
}
```

Quando l'operazione di riavvio è completa, il valore di `OperationState` è `REBOOT_COMPLETE`.

Riavviare un broker per un cluster Amazon MSK utilizzando l'API

Per riavviare un broker in un cluster utilizzando l'API, consulta. RebootBroker

Contrassegna un tag a un cluster Amazon MSK

Puoi assegnare i tuoi metadati sotto forma di tag a una risorsa Amazon MSK, ad esempio un cluster MSK. Un tag è una coppia chiave-valore che definisci per la flusso. L'utilizzo dei tag è un modo semplice ma efficace per gestire AWS le risorse e organizzare i dati, inclusi i dati di fatturazione.

Argomenti

- Nozioni di base sui tag per i cluster Amazon MSK

Nozioni di base sui tag per i cluster Amazon MSK

Puoi utilizzare l'API di Amazon MSK per completare le attività seguenti:

- Aggiunta di tag a una risorsa Amazon MSK.
- Elenco dei tag per una risorsa Amazon MSK.
- Rimozione dei tag da una risorsa Amazon MSK.

Puoi utilizzare i tag per categorizzare le risorse Amazon MSK. Ad esempio, puoi categorizzare i cluster Amazon MSK in base a scopo, proprietario o ambiente. Poiché definisci una chiave e un valore per ogni tag, puoi creare un set di categorie personalizzate per soddisfare esigenze specifiche. Ad esempio, puoi definire un set di tag che consente di monitorare i cluster in base al proprietario e all'applicazione associata.

Di seguito sono illustrati alcuni esempi di tag:

- `Project: Project name`
- `Owner: Name`
- `Purpose: Load testing`
- `Environment: Production`

Tieni traccia dei costi dei cluster Amazon MSK utilizzando i tag

Puoi utilizzare i tag per classificare e tenere traccia dei costi. AWS Quando applichi tag alle tue AWS risorse, inclusi i cluster Amazon MSK, il report sull'allocazione dei AWS costi include l'utilizzo e i costi aggregati per tag. Puoi organizzare i costi tra più servizi applicando tag che rappresentano categorie di business (come centri di costo, nomi di applicazioni o proprietari). Per ulteriori informazioni, consulta Utilizzo dei tag per l'allocazione dei costi ai fini dei report di fatturazione personalizzati nella AWS Billing User Guide (Guida per l'utente di Amazon API Gateway).

Limitazioni applicate ai tag

Ai tag in Amazon MSK si applicano le limitazioni seguenti.

Limitazioni di base

- Il numero massimo di tag per risorsa è 50.
- Per le chiavi e i valori dei tag viene fatta la distinzione tra maiuscole e minuscole.
- Non è possibile cambiare o modificare i tag di una risorsa eliminata.

Limitazioni applicate alle chiavi di tag

- Ogni chiave di tag deve essere univoca. Se aggiungi un tag con una chiave già in uso, il nuovo tag sovrascrive la coppia chiave-valore esistente.
- Una chiave di tag non può iniziare con `aws:` perché questo prefisso è riservato per l'utilizzo da parte di AWS. AWS crea tag con questo prefisso per tuo conto, ma non puoi modificarli o eliminarli.
- Le chiavi di tag devono avere una lunghezza compresa tra 1 e 128 caratteri Unicode.
- Le chiavi di tag devono contenere i seguenti caratteri: lettere Unicode, cifre, spazio e i seguenti caratteri speciali: `_ . / = + - @`.

Limitazioni applicate ai valori dei tag

- I valori dei tag devono avere una lunghezza compresa tra 0 e 255 caratteri Unicode.
- I valori dei tag possono essere vuoti. In caso contrario, devono contenere i seguenti caratteri: lettere Unicode, cifre, spazio e i seguenti caratteri speciali: `_ . / = + - @`.

Contrassegna le risorse utilizzando l'API Amazon MSK

Puoi utilizzare le operazioni seguenti per assegnare o rimuovere i tag da una risorsa Amazon MSK o per elencare il set di tag corrente per una risorsa:

- ListTagsForResource
- TagResource
- UntagResource

Migrazione a un cluster Amazon MSK

Il replicatore Amazon MSK può essere utilizzato per eseguire la migrazione dei cluster MSK. Consultare Cos'è il replicatore Amazon MSK?. In alternativa, è possibile utilizzare Apache

MirrorMaker 2.0 per migrare da un cluster non MSK a un cluster Amazon MSK. Per un esempio di come eseguire questa operazione, consulta la pagina Migrate an on-premises Apache Kafka cluster to Amazon MSK by using. MirrorMaker Per informazioni sull'uso MirrorMaker, consulta Mirroring dei dati tra i cluster nella documentazione di Apache Kafka. Ti consigliamo di eseguire la configurazione in una configurazione ad alta MirrorMaker disponibilità.

Una descrizione dei passaggi da completare quando si utilizza per eseguire la migrazione MirrorMaker a un cluster MSK

1. Creazione del cluster MSK di destinazione
2. Inizia MirrorMaker da un' EC2 istanza Amazon all'interno dello stesso Amazon VPC del cluster di destinazione.
3. Ispeziona il ritardo MirrorMaker .
4. Dopo aver MirrorMaker recuperato il ritardo, reindirizza produttori e consumatori al nuovo cluster utilizzando i broker bootstrap del cluster MSK.
5. Chiusura MirrorMaker.

Migrazione del cluster Apache Kafka ad Amazon MSK

Supponi di disporre di un cluster Apache Kafka denominato `CLUSTER_ONPREM`, popolato con argomenti e dati. Se desideri eseguire la migrazione di tale cluster in un nuovo cluster Amazon MSK denominato `CLUSTER_AWSMSK`, questa procedura fornisce una vista generale dei passaggi necessari.

Migrazione del cluster Apache Kafka esistente ad Amazon MSK

1. In `CLUSTER_AWSMSK`, creare tutti gli argomenti che desideri migrare.

   Non puoi utilizzarlo MirrorMaker per questo passaggio perché non ricrea automaticamente gli argomenti che desideri migrare con il giusto livello di replica. È possibile creare gli argomenti in Amazon MSK con gli stessi fattori di replica e numeri di partizioni che esistevano in `CLUSTER_ONPREM`. È possibile inoltre creare gli argomenti con diversi fattori di replica e numeri di partizioni.
2. Inizia MirrorMaker da un'istanza con accesso in lettura `CLUSTER_ONPREM` e accesso in scrittura. `CLUSTER_AWSMSK`
3. Eseguire il comando seguente per creare una copia speculare di tutti gli argomenti:

```
<path-to-your-kafka-installation>/bin/kafka-mirror-maker.sh --consumer.config
 config/mirrormaker-consumer.properties --producer.config config/mirrormaker-
producer.properties --whitelist '.*'
```

In questo comando, `config/mirrormaker-consumer.properties` punta a un broker bootstrap in `CLUSTER_ONPREM`; ad esempio, `bootstrap.servers=localhost:9092`. E `config/mirrormaker-producer.properties` indica un broker di bootstrap in CLUSTER_AWSMSK; ad esempio,. `bootstrap.servers=10.0.0.237:9092,10.0.2.196:9092,10.0.1.233:9092`

4. Continua a MirrorMaker funzionare in background e continua a utilizzare. `CLUSTER_ONPREM` MirrorMaker rispecchia tutti i nuovi dati.
5. Controlla lo stato di avanzamento del mirroring controllando il ritardo tra l'ultimo offset di ogni argomento e l'offset corrente da cui si sta consumando. MirrorMaker

   Ricorda che si MirrorMaker tratta semplicemente di utilizzare un consumatore e un produttore. Quindi, è possibile controllare il ritardo usando lo strumento `kafka-consumer-groups.sh`. Per trovare il nome del gruppo di consumatori, cercare all'interno del file `group.id` `mirrormaker-consumer.properties` e utilizzare il suo valore. Se tale chiave non esiste nel file, è possibile crearla. Ad esempio, impostare `group.id=mirrormaker-consumer-group`.
6. Dopo aver MirrorMaker finito di rispecchiare tutti gli argomenti, interrompete tutti i produttori e i consumatori, e poi smettete MirrorMaker. Quindi, reindirizzare i produttori e i consumatori al cluster `CLUSTER_AWSMSK` modificando i relativi valori dei broker bootstrap del produttore e del consumatore. Riavviare tutti i produttori e i consumatori su `CLUSTER_AWSMSK`.

Migrazione da un cluster Amazon MSK a un altro

È possibile utilizzare Apache MirrorMaker 2.0 per migrare da un cluster non MSK a un cluster MSK. Ad esempio, puoi eseguire la migrazione da una versione di Apache Kafka a un'altra. Per un esempio di come eseguire questa operazione, consulta la pagina Migrate an on-premises Apache Kafka cluster to Amazon MSK by using. MirrorMaker In alternativa, è possibile utilizzare il replicatore Amazon MSK per eseguire la migrazione dei cluster MSK. Per ulteriori informazioni sul replicatore Amazon MSK, consulta la pagina Cos'è il replicatore Amazon MSK?.

MirrorMaker 1.0 best practice

Questo elenco di best practice si applica alla MirrorMaker versione 1.0.

- Esegui MirrorMaker sul cluster di destinazione. In questo modo, se si verifica un problema di rete, i messaggi sono ancora disponibili nel cluster di origine. Se esegui MirrorMaker sul cluster di origine e gli eventi sono memorizzati nel buffer nel produttore e c'è un problema di rete, gli eventi potrebbero andare persi.
- Se è richiesta la crittografia dei dati in transito, eseguirla nel cluster di origine.
- Per i consumatori, impostare auto.commit.enabled=false
- Per i produttori, impostare
  - max.in.flight.requests.per.connection=1
  - retries=Int.Max_Value
  - acks=all
  - max.block.ms = Long.Max_Value
- Per un elevato throughput produttore:
  - Esegui il buffer di messaggi e compila batch di messaggi: tune buffer.memory, batch.size, linger.ms
  - Ottimizza i buffer dei socket: receive.buffer.bytes, send.buffer.bytes
- Per evitare la perdita di dati, disattiva il commit automatico all'origine, in modo che MirrorMaker possa controllare i commit, cosa che in genere esegue dopo aver ricevuto l'ack dal cluster di destinazione. Se il produttore ha acks=all e il cluster di destinazione ha impostato min.insync.replicas su più di 1, i messaggi vengono mantenuti su più di un broker nella destinazione prima che il consumatore esegua il commit dell'offset all'origine. MirrorMaker
- Se l'ordine è importante, puoi impostare i tentativi su 0. In alternativa, per un ambiente di produzione, imposta il numero massimo di connessioni in transito su 1 per garantire che non venga eseguito il commit dei batch inviati senza seguire un ordine se un batch non riesce a metà. In questo modo, ogni batch inviato viene ritentato finché il batch successivo non viene inviato. Se max.block.ms non è impostato sul valore massimo e se il buffer del produttore è pieno, potrebbe verificarsi una perdita di dati (a seconda di alcune delle altre impostazioni). Questo può bloccare e causare uno stato di congestione nel consumatore.
- Per elevato throughput
  - Incrementa buffer.memory.
  - Incrementa le dimensioni batch.
  - Ottimizza linger.ms per consentire il riempimento dei batch. Ciò consente inoltre una migliore compressione, meno utilizzo della larghezza di banda della rete e meno storage sul cluster. Questo comporta un aumento della conservazione.

    - Monitora l'utilizzo della CPU e della memoria.
- Per elevato throughput consumatore
    - Aumenta il numero di thread/consumatori per MirrorMaker processo: num.streams.
    - Aumenta il numero di MirrorMaker processi tra le macchine prima di aumentare i thread per consentire un'elevata disponibilità.
    - Aumenta il numero di MirrorMaker processi prima sulla stessa macchina e poi su macchine diverse (con lo stesso ID di gruppo).
    - Isola gli argomenti con una velocità effettiva molto elevata e utilizza istanze separate MirrorMaker .
- Per gestione e configurazione
    - Strumenti di gestione AWS CloudFormation dell'uso e della configurazione come Chef e Ansible.
    - Utilizza montaggi Amazon EFS per mantenere tutti i file di configurazione accessibili da tutte le EC2 istanze Amazon.
    - Usa i contenitori per scalare e gestire facilmente le istanze. MirrorMaker
- In genere, è necessario più di un consumatore per saturare un produttore. MirrorMaker Pertanto, configura più consumatori. Innanzitutto, configurali su macchine diverse per fornire elevata disponibilità. Quindi, dimensiona le singole macchine fino ad avere un consumatore per ogni partizione, con i consumatori distribuiti in modo uniforme tra le macchine.
- Per elevato throughput di inserimento e consegna, ottimizza i buffer di ricezione e invio perché le relative impostazione predefinite potrebbero essere troppo basse. Per ottenere le massime prestazioni, assicuratevi che il numero totale di stream (num.streams) corrisponda a tutte le partizioni degli argomenti che state tentando di copiare nel MirrorMaker cluster di destinazione.

Vantaggi di 2. MirrorMaker *

- Utilizza il framework e l'ecosistema Apache Kafka Connect.
- Rileva nuovi argomenti e partizioni.
- Sincronizza automaticamente la configurazione degli argomenti tra cluster.
- Supporta coppie di cluster "attiva/attiva", così come qualsiasi numero di cluster attivi.
- Fornisce nuove metriche, tra cui end-to-end la latenza di replica su più data center e cluster.
- Emette gli offset necessari per eseguire la migrazione dei consumatori tra cluster e fornisce strumenti per la traslazione dell'offset.

- Supporta un file di configurazione di alto livello per specificare più cluster e flussi di replica in un'unica posizione, rispetto alle proprietà produttore/consumatore di basso livello per ogni processo 1.*. MirrorMaker

Eliminare un cluster Amazon MSK Provisioned

> **Note**
>
> Se il cluster Amazon MSK di cui hai effettuato il provisioning ha una politica di auto-scaling, ti consigliamo di rimuovere la policy prima di eliminare il cluster. Per ulteriori informazioni, consulta Scalabilità automatica per i cluster Amazon MSK.

Argomenti

Eliminare un cluster Amazon MSK Provisioned utilizzando AWS Management Console

Questo processo descrive come eliminare un cluster Amazon MSK Provisioned utilizzando il. AWS Management Console Prima di eliminare un cluster MSK, assicurati di disporre di un backup di tutti i dati importanti memorizzati nel cluster e che non vi siano attività pianificate dipendenti dal cluster. Non è possibile annullare l'eliminazione di un cluster MSK.

1. Accedere a e aprire la console Amazon MSK da https://console.aws.amazon.com/msk/casa? AWS Management Console region=us-east-1#/home/.
2. Scegli il cluster MSK da eliminare selezionando la casella di controllo accanto ad esso.
3. Scegli Elimina e conferma l'eliminazione.

Eliminare un cluster Amazon MSK Provisioned utilizzando AWS CLI

Questo processo descrive come eliminare un cluster MSK Provisioned utilizzando. AWS CLI Prima di eliminare un cluster MSK, assicurati di disporre di un backup di tutti i dati importanti memorizzati nel cluster e che non vi siano attività pianificate dipendenti dal cluster. Non è possibile annullare l'eliminazione di un cluster MSK.

Esegui il comando seguente, sostituendolo *ClusterArn* con l'Amazon Resource Name (ARN) che hai ottenuto quando hai creato il cluster. Se non disponi dell'ARN per il cluster, puoi trovarlo elencando tutti i cluster. Per ulteriori informazioni, consulta the section called “Elenca i cluster”.

```
aws kafka delete-cluster --cluster-arn ClusterArn
```

Eliminare un cluster Amazon MSK Provisioned utilizzando l'API

L'API Amazon MSK ti consente di creare e gestire in modo programmatico il tuo cluster MSK Provisioned come parte di script di provisioning o distribuzione automatizzati dell'infrastruttura. Questo processo descrive come eliminare un cluster Amazon MSK Provisioned utilizzando l'API Amazon MSK. Prima di eliminare un cluster Amazon MSK, assicurati di disporre di un backup di tutti i dati importanti memorizzati nel cluster e che non vi siano attività pianificate dipendenti dal cluster. Non è possibile annullare l'eliminazione di un cluster MSK.

Per eliminare un cluster utilizzando l'API, vedi. DeleteCluster

Caratteristiche e concetti chiave di Amazon MSK

I cluster Amazon MSK Provisioned offrono un'ampia gamma di caratteristiche e funzionalità per aiutarti a ottimizzare le prestazioni del cluster e soddisfare le tue esigenze di streaming. Gli argomenti seguenti descrivono queste funzionalità in dettaglio.

- Il AWS Management Console
- La documentazione di riferimento all'API di Amazon MSK
- La documentazione di riferimento ai comandi della CLI di Amazon MSK

Argomenti

- Registrazione Amazon MSK
- Gestione dei metadati
- Risorse Amazon MSK
- Versioni di Apache Kafka
- Risolvi i problemi del tuo cluster Amazon MSK

Tipi di broker Amazon MSK

MSK Provisioned offre due tipi di broker: Standard ed Express. I broker standard offrono la massima flessibilità per configurare i cluster, mentre i broker Express offrono maggiore elasticità, velocità effettiva e resilienza e ease-of-use per l'esecuzione di applicazioni di streaming ad alte prestazioni. Consulta le sottosezioni seguenti per maggiori dettagli su ciascuna offerta. La tabella seguente evidenzia anche il confronto delle funzionalità chiave tra i broker Standard ed Express.

Confronto dei tipi di broker MSK Provisioned

Funzionalità	Broker standard	Broker espresso
Gestione dello storage	Gestita dal cliente (le funzionalità includono storage EBS, storage su più livelli, throughput di storage fornito, scalabilità automatica, avvisi sulla capacità di storage)	Completamente gestito da MSK
Istanze supportate	T3, M5, M7g	M7g
Considerazioni sul dimensionamento e sulla scalabilità	Throughput, connessioni, partizioni, archiviazione	Throughput, connessioni, partizioni
Scalabilità dei broker	Scalabilità verticale e orizzontale	Ridimensionamento verticale e orizzontale
Versioni Kafka	Per informazioni, consultare Versioni di Apache Kafka.	Inizia dalla versione 3.6
Configurazione di Apache Kafka	Più configurabile	Principalmente MSK è gestito per una maggiore resilienza

Funzionalità	Broker standard	Broker espresso
Sicurezza	Crittografia, Private/Public accesso, autenticazione e autorizzazione: IAM, SASL/SCRAM, mTLS, testo in chiaro, Kafka ACLs	Crittografia, Private/Public accesso, autenticazione e autorizzazione: IAM, SASL/SCRAM, mTLS, testo in chiaro, Kafka ACLs
Monitoraggio	CloudWatch, Monitoraggio aperto	CloudWatch, Monitoraggio aperto

> **Note**
>
> Non è possibile modificare un cluster MSK Provisioned da un tipo di broker Standard a un tipo di broker Express cambiando il tipo di broker utilizzando l'API MSK. È necessario creare un nuovo cluster con il tipo di broker desiderato (Standard o Express).

Argomenti

- Broker Amazon MSK Standard
- Broker Amazon MSK Express

Broker Amazon MSK Standard

I broker standard per MSK Provisioned offrono la massima flessibilità per configurare le prestazioni del cluster. È possibile scegliere tra un'ampia gamma di configurazioni di cluster per ottenere le caratteristiche di disponibilità, durabilità, velocità effettiva e latenza richieste per le applicazioni. È inoltre possibile fornire la capacità di storage e aumentarla in base alle esigenze. Amazon MSK gestisce la manutenzione hardware dei broker Standard e delle risorse di storage collegate, riparando automaticamente i problemi hardware che possono insorgere. Puoi trovare maggiori dettagli in questo documento su vari argomenti relativi ai broker Standard, inclusi argomenti sulla gestione dello storage, le configurazioni e la manutenzione.

Broker Amazon MSK Express

I broker Express per MSK Provisioned rendono Apache Kafka più semplice da gestire, più conveniente da eseguire su larga scala e più elastico grazie alla bassa latenza prevista. I

broker includono uno pay-as-you-go storage scalabile automaticamente e che non richiede dimensionamento, provisioning o monitoraggio proattivo. A seconda della dimensione dell'istanza selezionata, ogni nodo del broker può fornire un throughput fino a 3 volte superiore per broker, scalare fino a 20 volte più velocemente e ripristinare il 90% più velocemente rispetto ai broker Apache Kafka standard. I broker Express sono preconfigurati con le best practice predefinite di Amazon MSK e applicano le quote di throughput dei clienti per ridurre al minimo la contesa di risorse tra i clienti e le operazioni in background di Kafka.

Ecco alcuni fattori e funzionalità chiave da considerare quando si utilizzano i broker Express.

- Nessuna gestione dello storage: i broker Express eliminano la necessità di fornire o gestire qualsiasi risorsa di storage. Ottieni uno storage elastico pay-as-you-go, praticamente illimitato e completamente gestito. Per i casi d'uso con throughput elevato, non è necessario ragionare sulle interazioni tra istanze di elaborazione e volumi di storage e sui relativi colli di bottiglia in termini di throughput. Queste funzionalità semplificano la gestione dei cluster ed eliminano il sovraccarico operativo della gestione dello storage.
- Scalabilità più rapida: i broker Express consentono di scalare il cluster e spostare le partizioni fino a 20 volte più velocemente rispetto ai broker Standard. Questa funzionalità è fondamentale quando è necessario scalare il cluster per gestire i picchi di carico imminenti o scalare il cluster per ridurre i costi. Per maggiori dettagli sulla scalabilità del cluster, consulta le sezioni sull'espansione del cluster, la rimozione dei broker, la riassegnazione delle partizioni e LinkedInla configurazione del Cruise Control per il ribilanciamento.
- Throughput più elevato: i broker Express offrono un throughput fino a 3 volte superiore per broker rispetto ai broker Standard. Ad esempio, è possibile scrivere in sicurezza fino a 500 dati MBps con ogni broker Express di dimensioni pari a m7g.16xlarge rispetto ai 153,8 MBps del broker Standard equivalente (entrambi i numeri presuppongono un'allocazione della larghezza di banda sufficiente per le operazioni in background, come la replica e il ribilanciamento).
- Configurato per un'elevata resilienza: i broker Express offrono automaticamente diverse best practice per migliorare la resilienza del cluster. Queste includono protezioni sulle configurazioni critiche di Apache Kafka, quote di throughput e prenotazioni di capacità per operazioni in background e riparazioni non pianificate. Queste funzionalità rendono più sicura e semplice l'esecuzione di applicazioni Apache Kafka su larga scala. Consulta le sezioni relative Configurazioni del broker Express e Quota del broker Amazon MSK Express per maggiori dettagli.
- Nessuna finestra di manutenzione: non ci sono finestre di manutenzione per i broker Express. Amazon MSK aggiorna automaticamente l'hardware del cluster su base continuativa. Per maggiori dettagli, consulta Patching for Express brokers.

Informazioni aggiuntive sui broker Express

- I broker Express funzionano con Apache Kafka APIs, ma non supportano ancora completamente le API. KStreams
- I broker Express sono disponibili solo in una configurazione a 3. AZs
- I broker Express sono disponibili solo su istanze di dimensioni selezionate. Consulta i prezzi di Amazon MSK per l'elenco aggiornato.
- I broker Express sono supportati nelle versioni 3.6 e 3.8 di Apache Kafka.

Consulta questi blog

Per ulteriori informazioni sui broker MSK Express e per vedere un esempio reale di utilizzo dei broker Express, leggi i seguenti blog:

- Ti presentiamo Express Brokers for Amazon MSK per offrire un throughput elevato e una scalabilità più rapida per i tuoi cluster Kafka
- Express Brokers per Amazon MSK: scalabilità Kafka potenziata con prestazioni fino a 20 volte più veloci

  Questo blog dimostra come i broker Express:
  - Offrono un throughput più rapido, una scalabilità rapida e tempi di ripristino migliorati in caso di guasti
  - Elimina le complessità di gestione dello storage

Dimensioni dei broker Amazon MSK

Quando crei un cluster Amazon MSK Provisioned, specifichi la dimensione dei broker che desideri che abbia. A seconda del tipo di broker, Amazon MSK supporta le seguenti dimensioni di broker.

Dimensioni standard dei broker

- kafka.t3.small
- kafka.m5.large, kafka.m5.xlarge, kafka.m5.2xlarge, kafka.m5.4xlarge, kafka.m5.8xlarge, kafka.m5.12xlarge, kafka.m5.16xlarge, kafka.m5.24xlarge
- kafka.m7g.large, kafka.m7g.xlarge, kafka.m7g.2xlarge, kafka.m7g.4xlarge, kafka.m7g.8xlarge, kafka.m7g.12xlarge, kafka.m7g.16xlarge, kafka.m7g.16xlarge

Dimensioni dei broker Express

- express.m7g.large, express.m7g.xlarge, express.m7g.2xlarge, express.m7g.4xlarge, express.m7g.8xlarge, express.m7g.12xlarge, express.m7g.16xlarge, express.m7g.16xlarge

> **Note**
>
> Alcune dimensioni di broker potrebbero non essere disponibili in alcune regioni. AWS Consulta le tabelle dei prezzi aggiornate delle istanze Broker nella pagina dei prezzi di Amazon MSK per l'elenco più recente delle istanze disponibili per regione.

Altre note sulle dimensioni dei broker

- I broker M7g utilizzano processori AWS Graviton (processori personalizzati basati su ARM creati da Amazon Web Services). I broker M7g offrono un rapporto prezzo/prestazioni migliorato rispetto alle istanze M5 comparabili. I broker M7g consumano meno energia rispetto alle istanze M5 comparabili.
- Amazon MSK supporta i broker M7g su cluster MSK Provisioned con versioni 2.8.2 e 3.3.2 e successive di Kafka.
- I broker M7g e M5 offrono prestazioni di throughput di base più elevate rispetto ai broker T3 e sono consigliati per carichi di lavoro di produzione. I broker M7g e M5 possono anche avere più partizioni per broker rispetto ai broker T3. Usa i broker M7g o M5 se esegui carichi di lavoro di livello di produzione più grandi o richiedi un numero maggiore di partizioni. Per ulteriori informazioni sulle dimensioni delle istanze M7g e M5, consulta Amazon EC2 General Purpose Instances.
- I broker T3 possono utilizzare i crediti della CPU per incrementare temporaneamente le prestazioni. Utilizza i broker T3 per lo sviluppo a basso costo, se stai provando carichi di lavoro di streaming di piccole e medie dimensioni o se disponi di carichi di lavoro di streaming a throughput basso con picchi temporanei di throughput. Ti consigliamo di eseguire un proof-of-concept test per determinare se i broker T3 sono sufficienti per la produzione o per un carico di lavoro critico. Per ulteriori informazioni sulle dimensioni dei broker T3, consulta Amazon EC2 T3 Instances.

Per ulteriori informazioni su come scegliere le dimensioni dei broker, consulta. Le migliori pratiche per i broker Standard ed Express

Gestione dello storage per broker Standard

Amazon MSK offre funzionalità per aiutarti con la gestione dell'archiviazione sui tuoi cluster MSK.

> **Note**
>
> Con i broker Express, non è necessario fornire o gestire le risorse di archiviazione utilizzate per i dati. Ciò semplifica la gestione dei cluster ed elimina una delle cause più comuni dei problemi operativi con i cluster Apache Kafka. Inoltre, spendi meno in quanto non devi fornire capacità di storage inattiva e paghi solo per ciò che utilizzi.

Tipo di broker standard

Con i broker Standard puoi scegliere tra una varietà di opzioni e funzionalità di archiviazione. Amazon MSK offre funzionalità per aiutarti con la gestione dell'archiviazione sui tuoi cluster MSK.

Per informazioni sulla gestione del throughput, consulta. ???

Argomenti

- Storage su più livelli per broker Standard
- Espandi lo storage dei broker Amazon MSK Standard
- Gestisci il throughput di storage per i broker Standard in un cluster Amazon MSK

Storage su più livelli per broker Standard

L'archiviazione a più livelli è un livello di archiviazione a basso costo per Amazon MSK che si dimensiona fino a una capacità praticamente illimitata, rendendo conveniente la creazione di applicazioni di streaming di dati.

È possibile creare un cluster Amazon MSK configurato con un'archiviazione a più livelli che bilancia prestazioni e costi. Amazon MSK archivia i dati in streaming in un livello di archiviazione primario ottimizzato per le prestazioni fino a raggiungere i limiti di conservazione degli argomenti di Apache Kafka. Quindi, Amazon MSK sposta automaticamente i dati nel nuovo livello di archiviazione a basso costo.

Quando l'applicazione inizia a leggere i dati dall'archiviazione a più livelli, è possibile che i primi byte siano soggetti a un aumento della latenza di lettura. Quando inizi a leggere i dati rimanenti in sequenza dal livello a basso costo, le latenze dovrebbero essere simili a quelle del livello di

archiviazione primario. Non è necessario effettuare il provisioning di alcun tipo di archiviazione per l'archiviazione più livelli a basso costo o per gestire l'infrastruttura. È possibile archiviare qualsiasi quantità di dati e pagare solo per le risorse utilizzate. Questa funzionalità è compatibile con la funzionalità APIs introdotta in KIP-405: Kafka Tiered Storage.

Per informazioni sul dimensionamento, il monitoraggio e l'ottimizzazione del cluster di storage su più livelli MSK, consulta Best practice per l'esecuzione di carichi di lavoro di produzione utilizzando lo storage su più livelli Amazon MSK.

Di seguito sono elencate alcune caratteristiche dell'archiviazione a più livelli:

- È possibile dimensionare fino a una capacità di archiviazione praticamente illimitata. Non è necessario fare supposizioni su come dimensionare la propria infrastruttura Apache Kafka.
- È possibile mantenere i dati più a lungo negli argomenti di Apache Kafka o aumentare lo spazio di archiviazione degli argomenti senza la necessità di aumentare il numero di broker.
- Fornisce un buffer di sicurezza di maggiore durata per gestire ritardi imprevisti nell'elaborazione.
- Puoi rielaborare i vecchi dati nel loro esatto ordine di produzione con il codice di elaborazione dello stream esistente e Kafka. APIs
- Le partizioni si ribilanciano più velocemente perché i dati nell'archiviazione secondaria non richiedono la replica tra i dischi del broker.
- I dati tra i broker e l'archiviazione a più livelli si spostano all'interno del VPC e non viaggiano su Internet.
- Per connettersi a nuovi cluster con l'archiviazione a più livelli abilitata, un computer client può utilizzare lo stesso processo che utilizza per connettersi a un cluster senza l'archiviazione a più livelli abilitata. Consulta la sezione Creazione di un computer client.

Requisiti di storage su più livelli per i cluster Amazon MSK

- È necessario utilizzare la versione 3.0.0 o successiva del client Apache Kafka per creare un nuovo argomento con l'archiviazione a più livelli abilitata. Per trasferire un argomento esistente all'archiviazione a più livelli, puoi riconfigurare un computer client che utilizza una versione del client Kafka precedente alla 3.0.0 (la versione minima supportata di Apache Kafka è 2.8.2.tiered) per abilitare l'archiviazione a più livelli. Consultare Fase 4: creare un argomento nel cluster Amazon MSK.
- Il cluster Amazon MSK con storage su più livelli abilitato deve utilizzare la versione 3.6.0 o successiva o 2.8.2.tiered.

Vincoli e limitazioni dello storage su più livelli per i cluster Amazon MSK

L'archiviazione a più livelli presenta i seguenti vincoli e limitazioni:

- Assicurati che i client non siano configurati per `read_committed` la lettura da remote_tier in Amazon MSK, a meno che l'applicazione non utilizzi attivamente la funzionalità delle transazioni.
- Lo storage su più livelli non è disponibile nelle regioni AWS GovCloud (Stati Uniti).
- L'archiviazione a più livelli si applica solo ai cluster in modalità assegnata.
- Lo storage su più livelli non supporta la dimensione del broker t3.small.
- Il periodo di conservazione minimo nell'archiviazione a basso costo è di 3 giorni. Non è previsto un periodo minimo di conservazione per l'archiviazione primaria.
- L'archiviazione a più livelli non supporta le directory di log multipli su un broker (funzionalità relative a JBOD).
- Lo storage su più livelli non supporta argomenti compatti. Assicurati che cleanup.policy sia configurato solo su «DELETE» per tutti gli argomenti per cui è attivato lo storage su più livelli.
- Il cluster di archiviazione a più livelli non supporta la modifica della politica log.cleanup.policy per un argomento dopo la sua creazione.
- Lo storage su più livelli può essere disabilitato per singoli argomenti ma non per l'intero cluster. Una volta disattivata, l'archiviazione a più livelli non può essere riattivata per un argomento.
- Se utilizzi la versione 2.8.2.tiered di Amazon MSK, puoi migrare solo a un'altra versione di Apache Kafka supportata dallo storage su più livelli. Se non desideri continuare a utilizzare una versione supportata dallo storage su più livelli, crea un nuovo cluster MSK e migra i tuoi dati su di esso.
- Lo kafka-log-dirs strumento non è in grado di riportare le dimensioni dei dati di storage su più livelli. Lo strumento riporta solo la dimensione dei segmenti di log nell'archiviazione primaria.

Per informazioni sulle impostazioni e sui vincoli predefiniti, è necessario prestare attenzione quando si configura lo storage su più livelli a livello di argomento, vedere. Linee guida per la configurazione a livello di argomento dello storage su più livelli di Amazon MSK

Come vengono copiati i segmenti di log nello storage su più livelli per un argomento di Amazon MSK

Quando abiliti l'archiviazione a più livelli per un argomento nuovo o esistente, Apache Kafka copia i segmenti di log chiusi dall'archiviazione primaria all'archiviazione a più livelli.

- Apache Kafka copia solo i segmenti di log chiusi. Copia tutti i messaggi all'interno del segmento di log in un'archiviazione a più livelli.

- I segmenti attivi non sono idonei per l'archiviazione a più livelli. La dimensione del segmento di log (segment.bytes) o il tempo di distribuzione del segmento (segment.ms) controllano la velocità di chiusura dei segmenti e la velocità con cui, successivamente, Apache Kafka li copia nell'archiviazione a più livelli.

Le impostazioni di conservazione per un argomento con l'archiviazione a più livelli abilitata sono diverse dalle impostazioni per un argomento senza l'archiviazione a più livelli abilitata. Le seguenti regole disciplinano la conservazione dei messaggi negli argomenti con l'archiviazione a più livelli abilitata:

- È possibile definire la conservazione in Apache Kafka con due impostazioni: log.retention.ms (durata) e log.retention.bytes (dimensioni). Queste impostazioni determinano la durata e le dimensioni totali dei dati che Apache Kafka conserva nel cluster. Indipendentemente dal fatto che si abiliti o meno la modalità di archiviazione a più livelli, queste configurazioni vengono impostate a livello di cluster. È possibile sovrascrivere le impostazioni a livello di argomento con le configurazioni degli argomenti.
- Quando si abilita l'archiviazione a più livelli, è possibile specificare anche per quanto tempo il livello di archiviazione primaria ad alte prestazioni archivia i dati. Ad esempio, se un argomento ha un'impostazione di conservazione complessiva (log.retention.ms) di 7 giorni e una conservazione locale (local.retention.ms) di 12 ore, l'archiviazione primaria del cluster conserva i dati solo per le prime 12 ore. Il livello di archiviazione a basso costo conserva i dati per tutti i 7 giorni.
- Al log completo si applicano le normali impostazioni di conservazione. Ciò include le parti primarie e a più livelli.
- Le impostazioni local.retention.ms o local.retention.bytes controllano la conservazione dei messaggi nell'archiviazione primaria. Quando i dati hanno raggiunto le soglie di impostazione della conservazione dell'archiviazione primaria (local.retention.ms/bytes) su un log completo, Apache Kafka copia i dati nell'archiviazione primaria a più livelli. I dati sono quindi idonei alla scadenza.
- Quando Apache Kafka copia un messaggio in un segmento di log a più livelli, lo rimuove dal cluster in base alle impostazioni retention.ms o retention.bytes.

Esempio di scenario di storage su più livelli di Amazon MSK

Questo scenario illustra il comportamento di un argomento esistente che contiene messaggi nell'archiviazione primaria quando è abilitata l'archiviazione a più livelli. L'archiviazione a più livelli su questo argomento viene abilitata quando si imposta remote.storage.enable su `true`. In questo

esempio, retention.ms è impostato su 5 giorni e local.retention.ms è impostato su 2 giorni. Di seguito è riportata la sequenza di eventi alla scadenza di un segmento.

Ora T0: prima di abilitare l'archiviazione a più livelli.

Prima di abilitare l'archiviazione a più livelli per questo argomento, esistono due segmenti di log. Uno dei segmenti è attivo per una partizione di argomenti esistente 0.

Ora T1 (< 2 giorni): archiviazione a più livelli abilitata. Segmento 0 copiato nell'archiviazione a più livelli.

Dopo aver abilitato l'archiviazione a più livelli per questo argomento, Apache Kafka copia il segmento di log 0 nell'archiviazione a più livelli dopo che il segmento soddisfa le impostazioni di conservazione iniziali. Apache Kafka conserva anche la copia di archiviazione primaria del segmento 0. Il segmento 1 attivo non è ancora idoneo alla copia nell'archiviazione a più livelli. In questa sequenza temporale, Amazon MSK non applica ancora nessuna delle impostazioni di conservazione per nessuno dei messaggi nel segmento 0 e nel segmento 1. (conservazione locale). bytes/ms, retention.ms/bytes)

Ora T2: conservazione locale in vigore.

Dopo 2 giorni, le impostazioni di conservazione dell'archiviazione primaria hanno effetto per il segmento 0 che Apache Kafka ha copiato nell'archiviazione a più livelli. Ciò è determinato dall'impostazione di local.retention.ms su 2 giorni. Il segmento 0 ora scade dall'archiviazione primaria. Il segmento 1 è attivo, pertanto non è ancora idoneo né alla scadenza né a essere copiato nell'archiviazione a più livelli.

Ora T3: conservazione complessiva in vigore.

Dopo 5 giorni, le impostazioni di conservazione hanno effetto e Kafka cancella il segmento di log 0 e i messaggi associati dall'archiviazione a più livelli. Il segmento 1 non è ancora idoneo alla scadenza né può essere copiato nell'archiviazione a più livelli perché è attivo. Il segmento 1 non è ancora chiuso, quindi non è idoneo per la distribuzione dei segmenti.

Crea un cluster Amazon MSK con storage su più livelli con AWS Management Console

Questo processo descrive come creare un cluster Amazon MSK di storage su più livelli utilizzando. AWS Management Console

1. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/.
2. Scegli Create cluster (Crea cluster).
3. Scegli Creazione personalizzata per l'archiviazione a più livelli.
4. Specificare un nome per il cluster.
5. In Tipo di cluster, seleziona Assegnato.
6. Scegli la versione di Amazon Kafka che supporti l'archiviazione a più livelli e che desideri che Amazon MSK utilizzi per creare il cluster.
7. Specificate una dimensione del broker diversa da kafka.t3.small.
8. Specifica il numero di broker che devono essere creati da Amazon MSK in ogni zona di disponibilità. Il valore minimo è un broker per zona di disponibilità e il valore massimo è 30 broker per cluster.
9. Specifica il numero di zone in cui sono distribuiti i broker.
10. Specifica il numero di broker Apache Kafka implementati per zona.
11. Seleziona Opzioni di archiviazione. Ciò include l'archiviazione a più livelli e l'archiviazione EBS per abilitare la modalità di archiviazione a più livelli.
12. Segui i restanti passaggi nella procedura guidata di creazione dei cluster. Al termine, Archiviazione a più livelli e archiviazione EBS viene visualizzata come modalità di archiviazione del cluster nella vista Rivedi e crea.
13. Selezionare Creazione di un cluster.

Crea un cluster Amazon MSK con storage su più livelli con AWS CLI

Per abilitare l'archiviazione a più livelli su un cluster, crea il cluster con la versione e l'attributo di Apache Kafka corretti per l'archiviazione a più livelli. Segui l'esempio di codice sottostante. Inoltre, completa la procedura descritta nella sezione successiva per Crea un argomento su Kafka con lo storage su più livelli abilitato con AWS CLI.

Per un elenco completo degli attributi supportati per la creazione di cluster, consulta la sezione create-cluster.

```
aws kafka create-cluster \
 –cluster-name "MessagingCluster" \
 –broker-node-group-info file://brokernodegroupinfo.json \
 –number-of-broker-nodes 3 \
--kafka-version "3.6.0" \
```

```
--storage-mode "TIERED"
```

Crea un argomento su Kafka con lo storage su più livelli abilitato con AWS CLI

Per completare il processo avviato quando hai creato un cluster con l'archiviazione a più livelli abilitata, crea anche un argomento con l'archiviazione a più livelli abilitata con gli attributi dell'esempio di codice successivo. Gli attributi specifici per l'archiviazione a più livelli sono i seguenti:

- `local.retention.ms` (ad esempio, 10 minuti) per le impostazioni di conservazione basate sul tempo o `local.retention.bytes` per i limiti delle dimensioni dei segmenti di log.
- `remote.storage.enable` impostato su `true` per abilitare l'archiviazione a più livelli.

La configurazione seguente utilizza local.retention.ms, ma è possibile sostituire questo attributo con local.retention.bytes. Questo attributo controlla la quantità di tempo che può trascorrere o il numero di byte che Apache Kafka può copiare prima che il servizio copi i dati dall'archiviazione primaria a quella a più livelli. Per maggiori dettagli sugli attributi di configurazione supportati, consulta la sezione Configurazione a livello di argomento.

> **Note**
>
> È necessario utilizzare la versione 3.0.0 o successiva del client Apache Kafka. Queste versioni supportano un'impostazione chiamata `remote.storage.enable` solo in tali versioni client di `kafka-topics.sh`. Per abilitare l'archiviazione a più livelli su un argomento esistente che utilizza una versione precedente di Apache Kafka, consulta la sezione Abilitazione dello storage su più livelli su un argomento esistente di Amazon MSK.

```
bin/kafka-topics.sh --create --bootstrap-server $bs --replication-factor 2
 --partitions 6 --topic MSKTutorialTopic --config remote.storage.enable=true
 --config local.retention.ms=100000 --config retention.ms=604800000 --config
 segment.bytes=134217728
```

Abilitare e disabilitare lo storage su più livelli su un argomento esistente di Amazon MSK

Queste sezioni spiegano come abilitare e disabilitare l'archiviazione a più livelli su un argomento che hai già creato. Per creare un nuovo cluster e un argomento con l'archiviazione a più livelli abilitata, consulta la sezione Creazione di un cluster con archiviazione a più livelli tramite la AWS Management Console.

Abilitazione dello storage su più livelli su un argomento esistente di Amazon MSK

Per abilitare l'archiviazione a più livelli su un argomento esistente, utilizza la sintassi del comando `alter` nell'esempio seguente. Quando abiliti l'archiviazione a più livelli su un argomento esistente, non è necessario utilizzare una determinata versione del client Apache Kafka.

```
bin/kafka-configs.sh --bootstrap-server $bsrv --alter --entity-type topics
 --entity-name msk-ts-topic --add-config 'remote.storage.enable=true,
 local.retention.ms=604800000, retention.ms=15550000000'
```

Disabilitare lo storage su più livelli su un argomento esistente di Amazon MSK

Per disabilitare l'archiviazione a più livelli su un argomento esistente, utilizza la sintassi del comando `alter` nello stesso ordine con cui hai abilitato l'archiviazione a più livelli.

```
bin/kafka-configs.sh --bootstrap-server $bs --alter --entity-type topics --
entity-name MSKTutorialTopic --add-config 'remote.log.msk.disable.policy=Delete,
 remote.storage.enable=false'
```

> **Note**
>
> Quando si disabilita l'archiviazione a più livelli, si eliminano completamente i dati relativi all'argomento nell'archiviazione a più livelli. Apache Kafka conserva i dati dell'archiviazione primaria, ma applica comunque le regole di conservazione dell'archiviazione primaria in base a `local.retention.ms`. Una volta disabilitata l'archiviazione a più livelli su un argomento, non sarà possibile riabilitarla. Se desideri disabilitare l'archiviazione a più livelli su un argomento esistente, non è necessario utilizzare una determinata versione del client Apache Kafka.

Abilita lo storage su più livelli su un cluster Amazon MSK esistente tramite CLI AWS

> **Note**
>
> È possibile abilitare l'archiviazione a più livelli solo se la policy del cluster log.cleanup.policy è impostata su `delete`, poiché gli argomenti compatti non sono supportati nell'archiviazione a più livelli. Successivamente, puoi configurare la policy log.cleanup.policy di un singolo argomento su `compact` in modo che l'archiviazione a più livelli non sia abilitata su quel

particolare argomento. Per maggiori dettagli sugli attributi di configurazione supportati, consulta la sezione Configurazione a livello di argomento.

1. Aggiorna la versione di Kafka: le versioni dei cluster non sono semplici numeri interi. Per trovare la versione corrente del cluster, utilizzare l'`DescribeCluster`operazione o il comando `describe-cluster` AWS CLI. Una versione di esempio è `KTVPDKIKX0DER`.

```
aws kafka update-cluster-kafka-version --cluster-arn ClusterArn --current-version
 Current-Cluster-Version --target-kafka-version 3.6.0
```

2. Modifica la modalità di archiviazione del cluster. Nel seguente esempio di codice viene illustrato come modificare la modalità di archiviazione del cluster in `TIERED` tramite l'API `update-storage`.

```
aws kafka update-storage --current-version Current-Cluster-Version --cluster-arn
 Cluster-arn --storage-mode TIERED
```

Aggiorna lo storage su più livelli su un cluster Amazon MSK esistente utilizzando la console

Questo processo descrive come aggiornare un cluster Amazon MSK di storage su più livelli utilizzando il. AWS Management Console

Assicurati che la versione corrente di Apache Kafka del tuo cluster MSK sia la versione 2.8.2.tiered. Fai riferimento all'aggiornamento della versione di Apache Kafka se è necessario aggiornare il cluster MSK alla versione 2.8.2.tiered.

> **Note**
>
> È possibile abilitare l'archiviazione a più livelli solo se la policy del cluster log.cleanup.policy è impostata su `delete`, poiché gli argomenti compatti non sono supportati nell'archiviazione a più livelli. Successivamente, puoi configurare la policy log.cleanup.policy di un singolo argomento su `compact` in modo che l'archiviazione a più livelli non sia abilitata su quel particolare argomento. Per maggiori dettagli sugli attributi di configurazione supportati, consulta la sezione Configurazione a livello di argomento.

1. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/.

2. Vai alla pagina di riepilogo del cluster e scegli Proprietà.
3. Vai alla sezione Archiviazione e scegli Modifica modalità di archiviazione del cluster.
4. Scegli Archiviazione a più livelli e archiviazione EBS e Salva modifiche.

Espandi lo storage dei broker Amazon MSK Standard

È possibile aumentare la quantità di storage EBS per broker. Non è possibile ridurre lo storage.

I volumi di storage rimangono disponibili durante questa operazione di dimensionamento.

> **Important**
>
> Quando l'archiviazione viene dimensionata per un cluster MSK, l'archiviazione aggiuntiva viene resa disponibile immediatamente. Tuttavia, il cluster richiede un periodo di raffreddamento dopo ogni evento di dimensionamento dell'archiviazione. Amazon MSK utilizza questo periodo di raffreddamento per ottimizzare il cluster prima di un successivo nuovo dimensionamento. Questo periodo può variare da un minimo di 6 ore a più di 24 ore, a seconda delle dimensioni e dell'utilizzo dell'archiviazione del cluster e del traffico. Ciò è applicabile sia agli eventi di ridimensionamento automatico che al ridimensionamento manuale utilizzando l'UpdateBrokerStorageoperazione. Per informazioni sul corretto dimensionamento dell'archiviazione, consulta la sezione the section called “Best practice per broker standard”.

Puoi utilizzare l'archiviazione a più livelli per aumentare fino a quantità illimitate lo spazio di archiviazione per il broker. Per informazioni, consultare Storage su più livelli per broker Standard.

Argomenti

- Scalabilità automatica per i cluster Amazon MSK
- Ridimensionamento manuale per broker Standard

Scalabilità automatica per i cluster Amazon MSK

Per espandere automaticamente l'archiviazione del cluster in risposta a un maggiore utilizzo, puoi configurare una policy di dimensionamento automatico dell'applicazione per Amazon MSK. In una policy di dimensionamento automatico, si imposta l'utilizzo del disco di destinazione e la capacità di dimensionamento massima.

Prima di utilizzare il dimensionamento automatico per Amazon MSK, è consigliabile tenere in considerazione quanto segue:

- > **Important**
  >
  > Un'operazione di dimensionamento dell'archiviazione può avvenire solo una volta ogni sei ore.

  Ti consigliamo di iniziare con un volume di archiviazione della dimensione giusta per le tue esigenze di archiviazione. Per indicazioni sul corretto dimensionamento del cluster, consulta la pagina Dimensionamento corretto del cluster: numero di broker standard per cluster.
- Amazon MSK non riduce lo spazio di archiviazione del cluster in risposta a un utilizzo ridotto. Amazon MSK non supporta la riduzione delle dimensioni dei volumi di archiviazione. Se è necessario ridurre le dimensioni dell'archiviazione del cluster, è necessario migrare il cluster esistente in un cluster con un'archiviazione più piccola. Per ulteriori informazioni sulla migrazione di un cluster, consulta la pagina Migrazione al cluster Amazon MSK.
- Amazon MSK non supporta il dimensionamento automatico nelle regioni Asia Pacifico (Osaka-Locale) e Africa (Città del Capo).
- Quando associ una politica di auto-scaling al tuo cluster, Amazon Auto EC2 Scaling crea automaticamente un allarme CloudWatch Amazon per il tracciamento degli obiettivi. Se si elimina un cluster con una politica di auto-scaling, CloudWatch questo allarme persiste. Per eliminare l' CloudWatch allarme, è necessario rimuovere una politica di auto-scaling da un cluster prima di eliminare il cluster. Per ulteriori informazioni sul monitoraggio degli obiettivi, consulta le politiche di scalabilità di Target tracking per Amazon EC2 Auto Scaling nella Amazon Auto EC2 Scaling User Guide.

Argomenti

- Dettagli della politica di ridimensionamento automatico per Amazon MSK
- Configura il ridimensionamento automatico per il tuo cluster Amazon MSK

Dettagli della politica di ridimensionamento automatico per Amazon MSK

Una policy di dimensionamento automatico definisce i seguenti parametri predefiniti per il cluster:

- Obiettivo di utilizzo dell'archiviazione: la soglia di utilizzo dell'archiviazione utilizzata da Amazon MSK per attivare un'operazione di dimensionamento automatico. È possibile impostare l'obiettivo di utilizzo tra il 10% e l'80% della capacità di archiviazione corrente. Consigliamo di impostare l'obiettivo di utilizzo dell'archiviazione tra il 50% e il 60%.
- Capacità massima di archiviazione: il limite di scalabilità massimo che Amazon MSK può impostare per l'archiviazione del broker. È possibile impostare la capacità di archiviazione massima fino a 16 TiB per broker. Per ulteriori informazioni, consulta Quota di Amazon MSK.

Quando Amazon MSK rileva che il parametro `Maximum Disk Utilization` è uguale o superiore all'impostazione `Storage Utilization Target`, aumenta la capacità di archiviazione di una quantità pari al più grande tra due numeri: 10 GiB o il 10% dell'archiviazione corrente. Ad esempio, se hai 1.000 GiB, tale quantità è 100 GiB. Il servizio verifica l'utilizzo dell'archiviazione ogni minuto. Ulteriori operazioni di dimensionamento continuano ad aumentare l'archiviazione di una quantità pari al più grande tra due numeri: 10 GiB o il 10% dell'archiviazione corrente.

Per determinare se sono state eseguite operazioni di auto-scaling, utilizzare l'operazione. ListClusterOperations

Configura il ridimensionamento automatico per il tuo cluster Amazon MSK

Puoi utilizzare la console Amazon MSK, l'API Amazon MSK o implementare il ridimensionamento automatico AWS CloudFormation per lo storage. CloudFormation il supporto è disponibile tramite. Application Auto Scaling

> **Note**
>
> Non è possibile implementare il dimensionamento automatico al momento della creazione di un cluster. È necessario innanzitutto creare il cluster, quindi creare e abilitare una policy di dimensionamento automatico per il cluster. Tuttavia, puoi creare la policy mentre il servizio Amazon MSK crea il tuo cluster.

Argomenti

- Configura il ridimensionamento automatico utilizzando Amazon MSK AWS Management Console
- Configura il ridimensionamento automatico utilizzando la CLI
- Configura la scalabilità automatica per Amazon MSK utilizzando l'API

Configura il ridimensionamento automatico utilizzando Amazon MSK AWS Management Console

Questo processo descrive come utilizzare la console Amazon MSK per implementare la scalabilità automatica per lo storage.

1. Accedere a e aprire la console Amazon MSK da https://console.aws.amazon.com/msk/casa? AWS Management Console region=us-east-1#/home/.
2. Nell'elenco di cluster, scegli il tuo cluster. Questa operazione ti reindirizzerà a una pagina che elenca i dettagli sul cluster.
3. Nella sezione Dimensionamento automatico per l'archiviazione, scegli Configura.
4. Crea e assegna un nome a una policy di dimensionamento automatico. Specifica l'obiettivo di utilizzo dell'archiviazione, la capacità massima di archiviazione e il parametro obiettivo.
5. Scegli `Save changes`.

Quando salvi e abiliti la nuova policy, la policy diventa attiva per il cluster. Quando viene raggiunto l'obiettivo di utilizzo dell'archiviazione, Amazon MSK espande l'archiviazione del cluster.

Configura il ridimensionamento automatico utilizzando la CLI

Questo processo descrive come utilizzare la CLI di Amazon MSK per implementare la scalabilità automatica per lo storage.

1. Usa il RegisterScalableTargetcomando per registrare un obiettivo di utilizzo dello storage.
2. Usa il PutScalingPolicycomando per creare una politica di espansione automatica.

Configura la scalabilità automatica per Amazon MSK utilizzando l'API

Questo processo descrive come utilizzare l'API Amazon MSK per implementare la scalabilità automatica per lo storage.

1. Utilizza l' RegisterScalableTargetAPI per registrare un obiettivo di utilizzo dello storage.
2. Utilizza l' PutScalingPolicyAPI per creare una politica di espansione automatica.

Ridimensionamento manuale per broker Standard

Per incrementare lo storage, attendere che lo stato del cluster diventi `ACTIVE`. Il dimensionamento dell'archiviazione prevede un periodo di raffreddamento di almeno sei ore tra un evento e l'altro.

Anche se l'operazione rende immediatamente disponibile spazio di archiviazione aggiuntivo, il servizio esegue ottimizzazioni sul cluster che possono richiedere fino a 24 ore o più. La durata di queste ottimizzazioni è proporzionale alla dimensione dell'archiviazione.

Scalabilità dello spazio di archiviazione dei broker utilizzando il AWS Management Console

1. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/.
2. Scegli il cluster MSK per cui desideri aggiornare lo spazio di archiviazione del broker.
3. Nella sezione Archiviazione, scegli Modifica.
4. Specifica il volume di storage desiderato. La quantità di storage può essere solo aumentata, non diminuita.
5. Scegli Save changes (Salva modifiche).

Scalabilità dello storage dei broker utilizzando il AWS CLI

Esegui il comando seguente, sostituendolo *ClusterArn* con l'Amazon Resource Name (ARN) che hai ottenuto quando hai creato il cluster. Se non disponi dell'ARN per il cluster, puoi trovarlo elencando tutti i cluster. Per ulteriori informazioni, consulta the section called “Elenca i cluster”.

Sostituiscilo *Current-Cluster-Version* con la versione corrente del cluster.

> **Important**
>
> Le versioni del cluster non sono interi semplici. Per trovare la versione corrente del cluster, usa l'DescribeClusteroperazione o il comando AWS CLI describe-cluster. Una versione di esempio è `KTVPDKIKX0DER`.

Il *Target-Volume-in-GiB* parametro rappresenta la quantità di spazio di archiviazione che desideri assegnare a ciascun broker. È consentito aggiornare lo storage solo per tutti i broker. Non è possibile specificare singoli broker per i quali aggiornare lo storage. Il valore specificato *Target-Volume-in-GiB* deve essere un numero intero maggiore di 100 GiB. Lo storage per broker dopo l'operazione di aggiornamento non può superare 16384 GiB.

```
aws kafka update-broker-storage --cluster-arn ClusterArn --current-version Current-
Cluster-Version --target-broker-ebs-volume-info '{"KafkaBrokerNodeId": "All",
 "VolumeSizeGB": Target-Volume-in-GiB}'
```

Aumento delle dimensioni dello spazio di archiviazione del broker tramite l'API

Per aggiornare lo storage di un broker utilizzando l'API, consulta UpdateBrokerStorage.

Gestisci il throughput di storage per i broker Standard in un cluster Amazon MSK

Per informazioni su come fornire la velocità effettiva utilizzando la console, la CLI e l'API di Amazon MSK, consulta. ???

Argomenti

- Problemi di throughput del broker Amazon MSK e impostazioni di throughput massimo
- Misura il throughput di storage di un cluster Amazon MSK
- Valori di aggiornamento della configurazione per lo storage assegnato in un cluster Amazon MSK
- Esegui il provisioning del throughput di storage per i broker Standard in un cluster Amazon MSK

Problemi di throughput del broker Amazon MSK e impostazioni di throughput massimo

Le cause dei colli di bottiglia nel throughput dei broker sono molteplici: throughput di volume, throughput di rete da Amazon ad EC2 Amazon EBS e throughput di uscita Amazon. EC2 È possibile abilitare la velocità di trasmissione effettiva assegnata per regolare la velocità di trasmissione effettiva del volume. Tuttavia, le limitazioni del throughput dei broker possono essere causate dal throughput di rete da Amazon EC2 ad Amazon EBS e dal throughput di uscita di Amazon EC2.

La velocità EC2 di uscita di Amazon è influenzata dal numero di gruppi di consumatori e di consumatori per gruppo di consumatori. Inoltre, sia il throughput di rete EC2 da Amazon ad Amazon EBS che il throughput di EC2 uscita Amazon sono più elevati per broker di grandi dimensioni.

Per volumi di dimensioni pari o superiori a 10 GiB, è possibile assegnare una velocità di trasmissione effettiva dell'archiviazione pari o superiore a 250 MiB al secondo. L'impostazione predefinita è 250 MiB al secondo. Per effettuare il provisioning del throughput di storage, devi scegliere la dimensione del broker kafka.m5.4xlarge o superiore (oppure kafka.m7g.2xlarge o superiore) e puoi specificare il throughput massimo come mostrato nella tabella seguente.

dimensione del broker	Velocità di trasmissione effettiva massima (MiB/secondo)
kafka.m5.4xlarge	593

dimensione del broker	Velocità di trasmissione effettiva massima (MiB/secondo)
kafka.m5.8xlarge	850
kafka.m5.12xlarge	1000
kafka.m5.16xlarge	1000
kafka.m5.24xlarge	1000
kafka.m7 g. 2 x grande	312,5
kafka.m7g.4xlarge	625
kafka.m7g.8xlarge	1000
kafka.m7g. 12 x grande	1000
kafka.m7g. 16 x grande	1000

Misura il throughput di storage di un cluster Amazon MSK

È possibile utilizzare i parametri `VolumeReadBytes` e `VolumeWriteBytes` per misurare la velocità di trasmissione effettiva media di archiviazione di un cluster. La somma di questi due parametri fornisce la velocità di trasmissione effettiva media dell'archiviazione espressa in byte. Per ottenere la velocità di trasmissione effettiva media dell'archiviazione per un cluster, imposta questi due parametri su SUM e il periodo su 1 minuto, quindi utilizza la formula seguente.

```
Average storage throughput in MiB/s = (Sum(VolumeReadBytes) + Sum(VolumeWriteBytes)) /
 (60 * 1024 * 1024)
```

Per ulteriori informazioni sui parametri `VolumeReadBytes` e `VolumeWriteBytes`, consulta la sezione the section called “Monitoraggio del livello PER_BROKER”.

Valori di aggiornamento della configurazione per lo storage assegnato in un cluster Amazon MSK

Puoi aggiornare la configurazione di Amazon MSK prima o dopo aver attivato la velocità di trasmissione effettiva assegnata. Tuttavia, non vedrai la velocità di trasmissione effettiva

desiderata finché non eseguirai entrambe le operazioni: aggiornare il parametro di configurazione `num.replica.fetchers` e attivare la velocità di trasmissione effettiva assegnata.

Nella configurazione predefinita di Amazon MSK, `num.replica.fetchers` ha un valore di 2. Per aggiornare il `num.replica.fetchers`, puoi utilizzare i valori suggeriti dalla tabella seguente. Questi valori sono forniti a scopo indicativo. Si consiglia di modificare questi valori in base al proprio caso d'uso.

dimensione del broker	num.replica.fetchers
kafka.m5.4xlarge	4
kafka.m5.8xlarge	8
kafka.m5.12xlarge	14
kafka.m5.16xlarge	16
kafka.m5.24xlarge	16

La configurazione aggiornata potrebbe non avere effetto per un massimo di 24 ore e potrebbe richiedere più tempo quando un volume sorgente non è completamente utilizzato. Tuttavia, le prestazioni dei volumi di transizione sono almeno uguali a quelle dei volumi di archiviazione di origine durante il periodo di migrazione. Un volume da 1 TiB completamente utilizzato richiede in genere circa sei ore per migrare a una configurazione aggiornata.

Esegui il provisioning del throughput di storage per i broker Standard in un cluster Amazon MSK

I broker Amazon MSK mantengono i dati sui volumi di archiviazione. Lo storage I/O viene utilizzato quando i produttori scrivono nel cluster, quando i dati vengono replicati tra broker e quando i consumatori leggono dati che non sono in memoria. La velocità di trasmissione effettiva dell'archiviazione del volume è la velocità con cui i dati possono essere scritti e letti da un volume di archiviazione. La velocità di trasmissione effettiva dell'archiviazione assegnata è la capacità di specificare tale velocità per i broker del cluster.

È possibile specificare la velocità di throughput assegnata in MiB al secondo per i cluster i cui broker sono di dimensioni `kafka.m5.4xlarge` o superiori e se il volume di storage è pari o superiore a 10 GiB. È possibile specificare la velocità di trasmissione effettiva assegnata durante la creazione del

cluster. Inoltre, è possibile abilitare o disabilitare la velocità di trasmissione effettiva assegnata per un cluster che si trova nello stato `ACTIVE`.

Per informazioni sulla gestione della velocità effettiva, vedere. ???

Argomenti

- Effettua il provisioning del throughput di storage del cluster Amazon MSK utilizzando il AWS Management Console
- Effettua il provisioning del throughput di storage del cluster Amazon MSK utilizzando il AWS CLI
- Esegui il provisioning del throughput di storage durante la creazione di un cluster Amazon MSK utilizzando l'API

Effettua il provisioning del throughput di storage del cluster Amazon MSK utilizzando il AWS Management Console

Questo processo mostra un esempio di come è possibile utilizzare il AWS Management Console per creare un cluster Amazon MSK con throughput assegnato abilitato.

1. Accedere a e aprire la console Amazon MSK da https://console.aws.amazon.com/msk/casa? AWS Management Console region=us-east-1#/home/.
2. Scegli Create cluster (Crea cluster).
3. Scegli Creazione personalizzata.
4. Specificare un nome per il cluster.
5. Nella sezione Archiviazione, scegli Abilita.
6. Scegli un valore per la velocità di trasmissione effettiva dell'archiviazione per broker.
7. Scegli un VPC, zone e sottoreti, nonché un gruppo di sicurezza.
8. Scegli Next (Successivo).
9. Nella parte inferiore del passaggio Sicurezza, scegli Avanti.
10. Nella parte inferiore del passaggio Monitoraggio e tag, scegli Avanti.
11. Verifica le impostazioni del cluster, quindi scegli Crea cluster.

Effettua il provisioning del throughput di storage del cluster Amazon MSK utilizzando il AWS CLI

Questo processo mostra un esempio di come è possibile utilizzare il AWS CLI per creare un cluster con il throughput assegnato abilitato.

1. Copia il codice JSON seguente e incollalo in un file. Sostituisci i segnaposto dell'ID della sottorete IDs e del gruppo di sicurezza con i valori del tuo account. Assegna al file il nome `cluster-creation.json` e salvalo.

```
{
    "Provisioned": {
        "BrokerNodeGroupInfo":{
            "InstanceType":"kafka.m5.4xlarge",
            "ClientSubnets":[
                "Subnet-1-ID",
                "Subnet-2-ID"
            ],
            "SecurityGroups":[
                "Security-Group-ID"
            ],
            "StorageInfo": {
                "EbsStorageInfo": {
                    "VolumeSize": 10,
                    "ProvisionedThroughput": {
                        "Enabled": true,
                        "VolumeThroughput": 250
                    }
                }
            }
        },
        "EncryptionInfo": {
            "EncryptionInTransit": {
                "InCluster": false,
                "ClientBroker": "PLAINTEXT"
            }
        },
        "KafkaVersion":"2.8.1",
        "NumberOfBrokerNodes": 2
    },
    "ClusterName": "provisioned-throughput-example"
}
```

2. Esegui il AWS CLI comando seguente dalla directory in cui hai salvato il file JSON nel passaggio precedente.

```
aws kafka create-cluster-v2 --cli-input-json file://cluster-creation.json
```

Esegui il provisioning del throughput di storage durante la creazione di un cluster Amazon MSK utilizzando l'API

Per configurare il throughput di storage assegnato durante la creazione di un cluster, usa V2. CreateCluster

Sicurezza in Amazon MSK

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per ulteriori informazioni sui programmi di conformità applicabili a Streaming gestito da Amazon per Apache Kafka, consulta la pagina Amazon Web Services in Scope by Compliance Program.
- Sicurezza nel cloud: la responsabilità dell'utente è determinata dal AWS servizio che utilizza. Inoltre, sei responsabile anche di altri fattori, tra cui la riservatezza dei dati, i requisiti dell'azienda e le leggi e le normative applicabili.

La presente documentazione aiuta a comprendere come applicare il modello di responsabilità condivisa quando si utilizza Amazon MSK. Gli argomenti seguenti descrivono come configurare Amazon MSK per soddisfare gli obiettivi di sicurezza e conformità. Vengono inoltre fornite informazioni su come utilizzare altri servizi di Amazon Web Services che consentono di monitorare e proteggere le risorse Amazon MSK.

Argomenti

- Protezione dei dati in Streaming gestito da Amazon per Apache Kafka
- Autenticazione e autorizzazione per Amazon MSK APIs
- Autenticazione e autorizzazione per Apache Kafka APIs
- Modifica del gruppo di sicurezza di un cluster Amazon MSK
- Controlla l'accesso ai ZooKeeper nodi Apache nel tuo cluster Amazon MSK

- Convalida della conformità per Streaming gestito da Amazon per Apache Kafka
- Resilienza in Streaming gestito da Amazon per Apache Kafka
- Sicurezza dell'infrastruttura in Streaming gestito da Amazon per Apache Kafka

Protezione dei dati in Streaming gestito da Amazon per Apache Kafka

Il modello di responsabilità AWS condivisa modello si applica alla protezione dei dati in Amazon Managed Streaming for Apache Kafka. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i. Cloud AWS L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al Modello di responsabilità condivisa AWS e GDPR nel Blog sulla sicurezza AWS .

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta Lavorare con i CloudTrail percorsi nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-3.

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include

quando lavori con Amazon MSK o altro Servizi AWS utilizzando la console, l'API o AWS SDKs. AWS CLI I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

Argomenti

- Crittografia di Amazon MSK
- Inizia a usare la crittografia Amazon MSK
- Usa Amazon MSK APIs con endpoint VPC di interfaccia

Crittografia di Amazon MSK

Amazon MSK fornisce opzioni di crittografia dei dati che puoi utilizzare per soddisfare rigidi requisiti di gestione dei dati. I certificati utilizzati da Amazon MSK per la crittografia devono essere rinnovati ogni 13 mesi. Amazon MSK rinnova automaticamente questi certificati per tutti i cluster. I cluster Express broker rimangono attivi quando Amazon MSK avvia l'operazione di aggiornamento dei certificati. `ACTIVE` Per i cluster di broker standard, Amazon MSK imposta lo stato del cluster su `MAINTENANCE` quando avvia l'operazione di aggiornamento dei certificati. MSK lo riporta a quando l'aggiornamento è terminato. `ACTIVE` Mentre un cluster è in corso l'operazione di aggiornamento dei certificati, è possibile continuare a produrre e consumare dati, ma non è possibile eseguire alcuna operazione di aggiornamento su di esso.

Crittografia Amazon MSK a riposo

Amazon MSK si integra con AWS Key Management Service (KMS) per offrire una crittografia lato server trasparente. Amazon MSK esegue sempre la crittografia dei dati a riposo. Quando crei un cluster Amazon MSK, puoi specificare la AWS KMS key che desideri far utilizzare ad Amazon MSK per crittografare i dati a riposo. Se non specifichi una chiave KMS, Amazon MSK crea una chiave KMS gestita da Chiave gestita da AWS e la utilizza per tuo conto. Per ulteriori informazioni sulle chiavi KMS, consulta AWS KMS keys nella Guida per gli sviluppatori di AWS Key Management Service .

Crittografia Amazon MSK in transito

Amazon MSK utilizza TLS 1.2. Per impostazione predefinita, effettua la crittografia dei dati in transito tra i broker del cluster MSK. Puoi ignorare questa impostazione predefinita al momento della creazione del cluster.

Per la comunicazione tra client e broker, è necessario specificare una delle tre impostazioni seguenti:

- Consenti solo dati crittografati TLS. Si tratta dell'impostazione di default.
- Consenti dati non crittografati e dati crittografati TLS.
- Consenti solo dati non crittografati.

I broker Amazon MSK utilizzano certificati pubblici AWS Certificate Manager . Pertanto, qualsiasi truststore che considera attendibili gli Amazon Trust Services considera attendibili anche i certificati dei broker Amazon MSK.

Anche se consigliamo vivamente di abilitare la crittografia dei dati in transito, questa potrebbe aggiungere un sovraccarico aggiuntivo della CPU e alcuni millisecondi di latenza. Tuttavia, la maggior parte dei casi d'uso non è sensibile a queste differenze e l'entità dell'impatto dipende dalla configurazione del cluster, dei client e del profilo di utilizzo.

Inizia a usare la crittografia Amazon MSK

Durante la creazione di un cluster MSK, puoi specificare le impostazioni di crittografia in formato JSON. Di seguito è riportato un esempio.

```
{
   "EncryptionAtRest": {
       "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/abcdabcd-1234-
abcd-1234-abcd123e8e8e"
    },
   "EncryptionInTransit": {
        "InCluster": true,
        "ClientBroker": "TLS"
    }
}
```

Per `DataVolumeKMSKeyId`, puoi specificare una chiave gestita dal cliente o la Chiave gestita da AWS per MSK nel tuo account (`alias/aws/kafka`). Se non lo specifichi`EncryptionAtRest`, Amazon MSK crittografa comunque i tuoi dati inattivi in base a. Chiave gestita da AWS Per determinare la chiave utilizzata dal cluster, invia una richiesta `GET` o richiama l'operazione API `DescribeCluster`.

Per `EncryptionInTransit`, il valore predefinito di `InCluster` è true, ma puoi impostarlo su false se Amazon MSK non deve crittografare i dati durante il passaggio tra i broker.

Per specificare la modalità di crittografia per i dati in transito tra client e broker, imposta `ClientBroker` su uno di tre valori: `TLS`, `TLS_PLAINTEXT` o `PLAINTEXT`.

Argomenti

- Specificare le impostazioni di crittografia durante la creazione di un cluster Amazon MSK
- Prova la crittografia TLS di Amazon MSK

Specificare le impostazioni di crittografia durante la creazione di un cluster Amazon MSK

Questo processo descrive come specificare le impostazioni di crittografia durante la creazione di un cluster Amazon MSK.

Specificare le impostazioni di crittografia durante la creazione di un cluster

1. Salvare il contenuto dell'esempio precedente in un file e assegnare al file il nome desiderato. Ad esempio, chiamarlo `encryption-settings.json`.
2. Eseguire il comando `create-cluster` e utilizzare l'opzione `encryption-info` per puntare al file in cui il JSON di configurazione è stato salvato. Di seguito è riportato un esempio. Sostituisci *`{YOUR MSK VERSION}`* con una versione che corrisponda alla versione del client Apache Kafka. Per informazioni su come trovare la versione del cluster MSK in uso, consulta la pagina Determinazione della versione del cluster MSK. Tieni presente che l'utilizzo di una versione del client Apache Kafka diversa da quella del cluster MSK può causare il danneggiamento, la perdita dei dati e tempi di inattività di Apache Kafka.

```
aws kafka create-cluster --cluster-name "ExampleClusterName" --broker-node-group-
info file://brokernodegroupinfo.json --encryption-info file://encryptioninfo.json
 --kafka-version "{YOUR MSK VERSION}" --number-of-broker-nodes 3
```

Di seguito è riportato un esempio di una risposta corretta dopo l'esecuzione di questo comando.

```
{
    "ClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/SecondTLSTest/
abcdabcd-1234-abcd-1234-abcd123e8e8e",
    "ClusterName": "ExampleClusterName",
    "State": "CREATING"
}
```

Prova la crittografia TLS di Amazon MSK

Questo processo descrive come testare la crittografia TLS su Amazon MSK.

Per testare la crittografia TLS

1. Creare un computer client seguendo le linee guida in the section called “Crea una macchina client”.
2. Installare Apache Kafka sul computer client.
3. In questo esempio, utilizziamo il truststore JVM per comunicare con il cluster MSK. A tale scopo, creare innanzitutto una cartella denominata `/tmp` sul computer client. Quindi, passare alla cartella `bin` dell'installazione di Apache Kafka ed eseguire il seguente comando. (Il percorso JVM potrebbe essere diverso.)

   ```
   cp /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.201.b09-0.amzn2.x86_64/jre/lib/security/
   cacerts /tmp/kafka.client.truststore.jks
   ```

4. Sempre nella cartella `bin` dell'installazione di Apache Kafka sul computer client, creare un file di testo denominato `client.properties` con il seguente contenuto.

   ```
   security.protocol=SSL
   ssl.truststore.location=/tmp/kafka.client.truststore.jks
   ```

5. Esegui il comando seguente su un computer su cui è AWS CLI installato, sostituendolo *`clusterARN`* con l'ARN del tuo cluster.

   ```
   aws kafka get-bootstrap-brokers --cluster-arn clusterARN
   ```

   Se l'operazione riesce, il risultato sarà simile al seguente. Salvare questo risultato perché è necessario per la fase successiva.

   ```
   {
       "BootstrapBrokerStringTls": "a-1.example.g7oein.c2.kafka.us-
   east-1.amazonaws.com:0123,a-3.example.g7oein.c2.kafka.us-
   east-1.amazonaws.com:0123,a-2.example.g7oein.c2.kafka.us-east-1.amazonaws.com:0123"
   }
   ```

6. Esegui il comando seguente, sostituendolo *`BootstrapBrokerStringTls`* con uno degli endpoint del broker ottenuti nel passaggio precedente.

```
<path-to-your-kafka-installation>/bin/kafka-console-producer.sh --broker-
list BootstrapBrokerStringTls --producer.config client.properties --topic
 TLSTestTopic
```

7. Apri una nuova finestra di comando e connettiti allo stesso computer client. Quindi, esegui il comando seguente per creare un utente della console.

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBrokerStringTls --consumer.config client.properties --topic
 TLSTestTopic
```

8. Nella finestra del produttore, digita un messaggio di testo seguito da un invio e cerca lo stesso messaggio nella finestra del consumatore. Amazon MSK ha crittografato questo messaggio in transito.

Per ulteriori informazioni sulla configurazione dei client Apache Kafka per l'utilizzo di dati crittografati, consulta Configuring Kafka Clients.

Usa Amazon MSK APIs con endpoint VPC di interfaccia

Puoi utilizzare un endpoint VPC di interfaccia, alimentato da AWS PrivateLink, per impedire che il traffico tra Amazon VPC e Amazon MSK APIs lasci la rete Amazon. Gli endpoint VPC di interfaccia non richiedono un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione Direct Connect AWS . AWS PrivateLinkè una AWS tecnologia che consente la comunicazione privata tra AWS i servizi utilizzando un'interfaccia di rete elastica con private IPs nel tuo Amazon VPC. Per ulteriori informazioni, consulta Amazon Virtual Private Cloud and Interface VPC Endpoints ().AWS PrivateLink

Le tue applicazioni possono connettersi con Amazon MSK Provisioned e MSK Connect APIs utilizzando. AWS PrivateLink Per iniziare, crea un endpoint VPC di interfaccia per la tua API Amazon MSK per avviare il flusso di traffico da e verso le tue risorse Amazon VPC tramite l'endpoint VPC di interfaccia. Gli endpoint VPC con interfaccia FIPS sono disponibili per le regioni degli Stati Uniti. Per ulteriori informazioni, consulta Create an Interface Endpoint.

Utilizzando questa funzionalità, i client Apache Kafka possono recuperare dinamicamente le stringhe di connessione per connettersi alle risorse MSK Provisioned o MSK Connect senza dover attraversare Internet per recuperare le stringhe di connessione.

Quando crei un endpoint VPC di interfaccia, scegli uno dei seguenti endpoint con nome di servizio:

Per MSK Provisioned:

- com.amazonaws.region.kafka
- com.amazonaws.region.kafka-fips (compatibile con FIPS)

Dove region è il nome della tua regione. Scegliete questo nome di servizio per utilizzare MSK Provisioned APIs Compatible. Per ulteriori informazioni, vedere Operazioni nella versione 1.0/apireference/. https://docs.aws.amazon.com/msk/

Per MSK Connect:

- com.amazonaws.region.kafkaconnect

Dove regione è il nome della tua regione. Scegliete questo nome di servizio per lavorare con MSK Connect compatibile. APIs Per ulteriori informazioni, consulta Azioni nel riferimento all'API Amazon MSK Connect.

Per ulteriori informazioni, incluse step-by-step le istruzioni per creare un endpoint VPC di interfaccia, vedere Creazione di un endpoint di interfaccia nella Guida.AWS PrivateLink

Controlla l'accesso agli endpoint VPC per Amazon MSK Provisioned o MSK Connect APIs

Le policy degli endpoint VPC consentono di controllare l'accesso allegando una policy a un endpoint VPC o utilizzando campi aggiuntivi in una policy allegata a un utente, gruppo o ruolo IAM per limitare l'accesso solo tramite l'endpoint VPC specificato. Utilizzare la politica di esempio appropriata per definire le autorizzazioni di accesso per il servizio MSK Provisioned o MSK Connect.

Se non colleghi una policy durante la creazione di un endpoint, Amazon VPC collega una policy predefinita che consente l'accesso completo al servizio. Una policy endpoint non esclude né sostituisce policy IAM basate sull'identità o policy specifiche del servizio. Si tratta di una policy separata per controllare l'accesso dall'endpoint al servizio specificato.

Per ulteriori informazioni, consulta Controllare l'accesso ai servizi con gli endpoint VPC nella Guida.AWS PrivateLink

MSK Provisioned — VPC policy example

Accesso in sola lettura

Questa policy di esempio può essere associata a un endpoint VPC. Per ulteriori informazioni, consulta Controllo dell'accesso alle risorse VPC di Amazon. Limita le azioni solo a elencare e descrivere le operazioni tramite l'endpoint VPC a cui è collegato.

```
{
  "Statement": [
    {
      "Sid": "MSKReadOnly",
      "Principal": "*",
      "Action": [
        "kafka:List*",
        "kafka:Describe*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

MSK Provisioned — Esempio di policy per gli endpoint VPC

Limita l'accesso a un cluster MSK specifico

Questa policy di esempio può essere associata a un endpoint VPC. Limita l'accesso a uno specifico cluster Kafka tramite l'endpoint VPC a cui è collegato.

```
{
  "Statement": [
    {
      "Sid": "AccessToSpecificCluster",
      "Principal": "*",
      "Action": "kafka:*",
      "Effect": "Allow",
      "Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/MyCluster"
    }
  ]
}
```

MSK Connect — VPC endpoint policy example

Elenca i connettori e crea un nuovo connettore

Di seguito è riportato un esempio di policy sugli endpoint per MSK Connect. Questa politica consente al ruolo specificato di elencare i connettori e creare un nuovo connettore.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "MSKConnectPermissions",
            "Effect": "Allow",
            "Action": [
                "kafkaconnect:ListConnectors",
                "kafkaconnect:CreateConnector"
            ],
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:role/<ExampleRole>"
                ]
            }
        }
    ]
}
```

MSK Connect — Esempio di policy per gli endpoint VPC

Consente solo le richieste provenienti da un indirizzo IP specifico nel VPC specificato

L'esempio seguente mostra una policy che consente l'esito positivo solo delle richieste provenienti da un indirizzo IP specificato nel VPC specificato. Le richieste provenienti da altri indirizzi IP avranno esito negativo.

```
{
    "Statement": [
        {
            "Action": "kafkaconnect:*",
            "Effect": "Allow",
            "Principal": "*",
            "Resource": "*",
            "Condition": {
```

```
                "IpAddress": {
                    "aws:VpcSourceIp": "192.0.2.123"
                },
        "StringEquals": {
                    "aws:SourceVpc": "vpc-555555555555"
                }
            }
        }
    ]
}
```

Autenticazione e autorizzazione per Amazon MSK APIs

AWS Identity and Access Management (IAM) è uno strumento Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle risorse. AWS Gli amministratori IAM controllano chi può essere autenticato (accesso effettuato) e autorizzato (dotato di autorizzazioni) per utilizzare le risorse Amazon MSK. IAM è un software Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

- Funzionamento di Amazon MSK con IAM
- Esempi di policy basate sull'identità per Amazon MSK
- Ruoli collegati ai servizi per Amazon MSK
- AWS politiche gestite per Amazon MSK
- Risolvi i problemi relativi all'identità e all'accesso ad Amazon MSK

Funzionamento di Amazon MSK con IAM

Prima di utilizzare IAM per gestire l'accesso ad Amazon MSK, è necessario comprendere quali funzioni IAM sono disponibili per l'uso con Amazon MSK. Per avere una visione di alto livello di come Amazon MSK e altri AWS servizi funzionano con IAM, consulta AWS Services That Work with IAM nella IAM User Guide.

Argomenti

- Policy basate sull'identità di Amazon MSK
- Policy basate sulle risorse di Amazon MSK
- Autorizzazione basata sui tag Amazon MSK

- Ruoli IAM di Amazon MSK

Policy basate sull'identità di Amazon MSK

Con le policy basate su identità di IAM, è possibile specificare quali azioni e risorse sono consentite o rifiutate, nonché le condizioni in base alle quali le azioni sono consentite o rifiutate. Amazon MSK supporta operazioni, risorse e chiavi di condizione specifiche. Per informazioni su tutti gli elementi utilizzati in una policy JSON, consulta Documentazione di riferimento degli elementi delle policy JSON IAM nella Guida per l'utente IAM.

Azioni per le politiche basate sull'identità di Amazon MSK

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Action`di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le operazioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Le operazioni delle policy in Amazon MSK utilizzano il seguente prefisso prima dell'operazione: `kafka:`. Ad esempio, per concedere a qualcuno l'autorizzazione per descrivere un cluster MSK con l'operazione API `DescribeCluster` di Amazon MSK, includi l'operazione `kafka:DescribeCluster` nella policy. Le istruzioni della policy devono includere un elemento `Action` o `NotAction`. Amazon MSK definisce un proprio set di operazioni che descrivono le attività che puoi eseguire con quel servizio.

Per specificare più azioni in una sola istruzione, separa ciascuna di esse con una virgola come mostrato di seguito:

```
"Action": ["kafka:action1", "kafka:action2"]
```

È possibile specificare più azioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le azioni che iniziano con la parola `Describe`, includi la seguente azione:

```
"Action": "kafka:Describe*"
```

Per visualizzare un elenco delle operazioni di Amazon MSK, consulta la pagina Actions, resources, and condition keys for Amazon Managed Streaming for Apache Kafka nella Guida per l'utente di IAM.

Risorse per le politiche basate sull'identità di Amazon MSK

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource`della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Le istruzioni devono includere un elemento `Resource`o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). È possibile eseguire questa operazione per operazioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

La risorsa istanza di Amazon MSK dispone del seguente ARN:

```
arn:${Partition}:kafka:${Region}:${Account}:cluster/${ClusterName}/${UUID}
```

Per ulteriori informazioni sul formato di ARNs, consulta Amazon Resource Names (ARNs) e AWS Service Namespaces.

Ad esempio, per specificare l'istanza `CustomerMessages` nell'istruzione, utilizza il seguente ARN:

```
"Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/CustomerMessages/abcd1234-
abcd-dcba-4321-a1b2abcd9f9f-2"
```

Per specificare tutti le istanze che appartengono ad un account specifico, utilizza il carattere jolly (*):

```
"Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/*"
```

Alcune operazioni di Amazon MSK, ad esempio quelle per la creazione delle risorse, non possono essere eseguite su una risorsa specifica. In questi casi, è necessario utilizzare il carattere jolly (*).

```
"Resource": "*"
```

Per specificare più risorse in una singola istruzione, separale con virgole. ARNs

```
"Resource": ["resource1", "resource2"]
```

Per visualizzare un elenco dei tipi di risorse Amazon MSK e relativi ARNs, consulta Resources Defined by Amazon Managed Streaming for Apache Kafka nella IAM User Guide. Per informazioni sulle operazioni con cui è possibile specificare l'ARN di ogni risorsa, consulta la pagina Actions Defined by Amazon Managed Service for Apache Kafka.

Chiavi di condizione per le politiche basate sull'identità di Amazon MSK

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Condition`(o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition`è facoltativo. È possibile compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition`in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND`logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. `OR` Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

È possibile anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, è possibile autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Amazon MSK definisce il proprio set di chiavi di condizione e supporta anche l'utilizzo di alcune chiavi di condizione globali. Per vedere tutte le chiavi di condizione AWS globali, consulta AWS Global Condition Context Keys nella Guida per l'utente IAM.

Per visualizzare un elenco delle chiavi di condizione di Amazon MSK, consulta la pagina Condition Keys for Amazon Managed Streaming for Apache Kafka nella Guida per l'utente di IAM. Per

informazioni sulle operazioni e le risorse con le quali è possibile utilizzare una chiave di condizione, consulta la pagina Actions Defined by Amazon Managed Service for Apache Kafka.

Esempi di policy basate sull'identità di Amazon MSK

Per visualizzare degli esempi di policy basate sull'identità di Amazon MSK, consulta la pagina Esempi di policy basate sull'identità per Amazon MSK.

Policy basate sulle risorse di Amazon MSK

Amazon MSK supporta una policy del cluster (nota anche come policy basata sulle risorse) da utilizzare con i cluster Amazon MSK. Puoi utilizzare una policy del cluster per definire quali principali IAM dispongono delle autorizzazioni multi-account per configurare la connettività privata al tuo cluster Amazon MSK. In combinazione con l'autenticazione del client IAM, puoi utilizzare la policy del cluster anche per definire in modo granulare le autorizzazioni del piano dati Kafka per i client che si connettono.

Per visualizzare un esempio di come configurare una policy del cluster, consulta la sezione Passaggio 2: collegamento di una policy del cluster al cluster MSK.

Autorizzazione basata sui tag Amazon MSK

È possibile associare tag ai cluster Amazon MSK. Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `kafka:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* o `aws:TagKeys`. Per informazioni sull'etichettatura delle risorse Amazon MSK, consulta. the section called “Assegna un tag a un cluster”

Puoi controllare l'accesso al cluster solo con l'aiuto dei tag. Per etichettare argomenti e gruppi di consumatori, devi aggiungere una dichiarazione separata nelle tue politiche senza tag.

Per visualizzare un esempio di politica basata sull'identità per limitare l'accesso a un cluster in base ai tag di quel cluster, vedi. Accesso ai cluster Amazon MSK in base ai tag

Nella policy basata sull'identità, puoi utilizzare le condizioni per controllare l'accesso alle risorse Amazon MSK in base ai tag. L'esempio seguente mostra una policy che consente a un utente di descrivere il cluster, ottenere i relativi broker bootstrap, elencare i relativi nodi broker, aggiornarlo ed eliminarlo. Tuttavia, questa politica concede l'autorizzazione solo se il tag del cluster `Owner` ha il valore di quell'utente. `username` La seconda dichiarazione della seguente politica consente l'accesso agli argomenti del cluster. La prima dichiarazione di questa politica non autorizza l'accesso ad alcun argomento.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AccessClusterIfOwner",
      "Effect": "Allow",
      "Action": [
        "kafka:Describe*",
        "kafka:Get*",
        "kafka:List*",
        "kafka:Update*",
        "kafka:Delete*"
      ],
      "Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Owner": "${aws:username}"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "kafka-cluster:*Topic*",
        "kafka-cluster:WriteData",
        "kafka-cluster:ReadData"
      ],
      "Resource": [
        "arn:aws:kafka:us-east-1:123456789012:topic/*"
      ]
    }
  ]
}
```

Ruoli IAM di Amazon MSK

Un ruolo IAM è un'entità all'interno dell'account Amazon Web Services che dispone di autorizzazioni specifiche.

Utilizzo delle credenziali temporanee con Amazon MSK

È possibile utilizzare credenziali temporanee per effettuare l'accesso con la federazione, assumere un ruolo IAM o un ruolo multi-account. È possibile ottenere credenziali di sicurezza temporanee chiamando operazioni AWS STS API come AssumeRoleo. GetFederationToken

Amazon MSK supporta l'uso delle credenziali temporanee.

Ruoli collegati ai servizi

I ruoli collegati ai servizi permettono ad Amazon Web Services di accedere a risorse in altri servizi per completare un'operazione a tuo nome. I ruoli collegati ai servizi sono visualizzati nell'account IAM e sono di proprietà del servizio. Un amministratore può visualizzare, ma non modificare le autorizzazioni dei ruoli collegati ai servizi.

Amazon MSK supporta i ruoli collegati ai servizi. Per maggiori dettagli su come creare e gestire i ruoli collegati ai servizi di Amazon MSK, consulta la pagina the section called “Ruoli collegati ai servizi”.

Esempi di policy basate sull'identità per Amazon MSK

Per impostazione predefinita, gli utenti e i ruoli IAM non sono autorizzati a eseguire le operazioni API di Amazon MSK. Un amministratore deve creare le policy IAM che concedono a utenti e ruoli l'autorizzazione per eseguire operazioni API specifiche sulle risorse specificate di cui hanno bisogno. L'amministratore deve quindi allegare queste policy a utenti o IAM che richiedono tali autorizzazioni.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy nella scheda JSON nella Guida per l'utente IAM.

Argomenti

- Best practice delle policy
- Consentire agli utenti di visualizzare le loro autorizzazioni
- Accesso a un cluster Amazon MSK
- Accesso ai cluster Amazon MSK in base ai tag

Best practice delle policy

Le policy basate sull'identità determinano se qualcuno può creare, accedere o eliminare risorse Amazon MSK all'interno dell'account. Queste azioni possono comportare costi aggiuntivi per

l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche AWS gestite che concedono le autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta Policy gestite da AWSo Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegio minimo: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse è possibile aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle policy per il Sistema di analisi degli accessi IAM nella Guida per l'utente IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta Protezione dell'accesso API con MFA nella Guida per l'utente IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono cpllegate alla relativa identità utente. Questa policy include le autorizzazioni per completare questa azione sulla console o utilizzando programmaticamente l'API o. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Accesso a un cluster Amazon MSK

In questo esempio, si desidera concedere a un utente IAM nell'account Amazon Web Services l'accesso a uno dei cluster, `purchaseQueriesCluster`. Questa policy consente all'utente di descrivere il cluster, ottenere i broker bootstrap, elencare i nodi broker e aggiornarlo.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"UpdateCluster",
         "Effect":"Allow",
         "Action":[
            "kafka:Describe*",
            "kafka:Get*",
            "kafka:List*",
            "kafka:Update*"
         ],
         "Resource":"arn:aws:kafka:us-east-1:012345678012:cluster/
purchaseQueriesCluster/abcdefab-1234-abcd-5678-cdef0123ab01-2"
      }
   ]
}
```

Accesso ai cluster Amazon MSK in base ai tag

Nella policy basata sull'identità, puoi utilizzare le condizioni per controllare l'accesso alle risorse Amazon MSK in base ai tag. In questo esempio viene illustrato come creare una policy che consente all'utente di descrivere il cluster, ottenere i broker bootstrap, elencare i nodi broker, aggiornarlo ed eliminarlo. Tuttavia, l'autorizzazione viene concessa solo se il valore del tag di cluster `Owner` è quello del nome utente.

JSON

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

```
      "Sid": "AccessClusterIfOwner",
      "Effect": "Allow",
      "Action": [
        "kafka:Describe*",
        "kafka:Get*",
        "kafka:List*",
        "kafka:Update*",
        "kafka:Delete*"
      ],
      "Resource": "arn:aws:kafka:us-east-1:012345678012:cluster/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Owner": "${aws:username}"
        }
      }
    }
  ]
}
```

Puoi allegare questa policy agli utenti IAM nel tuo account. Se un utente denominato `richard-roe` tenta di aggiornare un cluster MSK, al cluster deve essere applicato il tag `0wner=richard-roe` o `owner=richard-roe`. In caso contrario, gli viene negato l'accesso. La chiave di tag di condizione `0wner` corrisponde a `0wner` e `owner` perché i nomi delle chiavi di condizione non effettuano la distinzione tra maiuscole e minuscole. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente di IAM.

Ruoli collegati ai servizi per Amazon MSK

Amazon MSK utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato ai servizi è un tipo di ruolo IAM univoco collegato direttamente ad Amazon MSK. I ruoli collegati ai servizi sono predefiniti da Amazon MSK e includono tutte le autorizzazioni richieste dal servizio per chiamare altri AWS servizi per tuo conto.

Un ruolo collegato ai servizi semplifica la configurazione di Amazon MSK perché consente di evitare l'aggiunta manuale delle autorizzazioni necessarie. Amazon MSK definisce le autorizzazioni dei ruoli collegati ai servizi. Se non diversamente definito, solo Amazon MSK può assumere i suoi ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consulta la pagina Amazon Web Services That Work with IAM e cerca i servizi che riportano Sì nella colonna Ruolo

collegato ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Argomenti

- Autorizzazioni del ruolo collegato ai servizi per Amazon MSK
- Crea un ruolo collegato ai servizi per Amazon MSK
- Modificare un ruolo collegato al servizio per Amazon MSK
- Regioni supportate per i ruoli collegati ai servizi di Amazon MSK

Autorizzazioni del ruolo collegato ai servizi per Amazon MSK

Amazon MSK usa il ruolo collegato ai servizi denominato AWSServiceRoleForKafka. Amazon MSK utilizza questo ruolo per accedere alle risorse ed eseguire operazioni come:

- `*NetworkInterface`: crea e gestisci interfacce di rete nell'account cliente che rendano i broker del cluster accessibili ai client nel VPC del cliente.
- `*VpcEndpoints`— gestire gli endpoint VPC nell'account cliente che rendono i broker di cluster accessibili ai clienti nel VPC del cliente utilizzando. AWS PrivateLink Amazon MSK utilizza le autorizzazioni per `DescribeVpcEndpoints`, `ModifyVpcEndpoint` e `DeleteVpcEndpoints`.
- `secretsmanager`— gestisci le credenziali dei clienti con. AWS Secrets Manager
- `GetCertificateAuthorityCertificate`: recupera il certificato per la tua autorità di certificazione privata.

Questo ruolo collegato ai servizi è collegato alle seguenti policy gestite: `KafkaServiceRolePolicy`. Per gli aggiornamenti a questa policy, consulta KafkaServiceRolePolicy.

Ai fini dell'assunzione del ruolo, il ruolo collegato ai servizi AWSServiceRoleForKafka considera attendibili i seguenti servizi:

- `kafka.amazonaws.com`

La policy delle autorizzazioni del ruolo consente ad Amazon MSK di completare le seguenti operazioni sulle risorse.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "ec2:CreateNetworkInterface",
    "ec2:DescribeNetworkInterfaces",
    "ec2:CreateNetworkInterfacePermission",
    "ec2:AttachNetworkInterface",
    "ec2:DeleteNetworkInterface",
    "ec2:DetachNetworkInterface",
    "ec2:DescribeVpcEndpoints",
    "acm-pca:GetCertificateAuthorityCertificate",
    "secretsmanager:ListSecrets"
   ],
   "Resource": "*"
  },
  {
   "Effect": "Allow",
   "Action": [
    "ec2:ModifyVpcEndpoint"
   ],
   "Resource": "arn:*:ec2:*:*:subnet/*"
  },
  {
   "Effect": "Allow",
   "Action": [
    "ec2:DeleteVpcEndpoints",
    "ec2:ModifyVpcEndpoint"
   ],
   "Resource": "arn:*:ec2:*:*:vpc-endpoint/*",
   "Condition": {
    "StringEquals": {
     "ec2:ResourceTag/AWSMSKManaged": "true"
    },
    "StringLike": {
     "ec2:ResourceTag/ClusterArn": "*"
    }
   }
  },
```

```
  {
   "Effect": "Allow",
   "Action": [
    "secretsmanager:GetResourcePolicy",
    "secretsmanager:PutResourcePolicy",
    "secretsmanager:DeleteResourcePolicy",
    "secretsmanager:DescribeSecret"
   ],
   "Resource": "*",
   "Condition": {
    "ArnLike": {
     "secretsmanager:SecretId": "arn:*:secretsmanager:*:*:secret:AmazonMSK_*"
    }
   }
  }
 ]
}
```

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato ai servizi devi configurare le relative autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Crea un ruolo collegato ai servizi per Amazon MSK

Non è necessario creare manualmente un ruolo collegato ai servizi. Quando crei un cluster Amazon MSK nell'API AWS Management Console, nell'o nell' AWS API AWS CLI, Amazon MSK crea automaticamente il ruolo collegato al servizio.

Se elimini questo ruolo collegato ai servizi, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando si crea un cluster Amazon MSK, Amazon MSK crea di nuovo automaticamente il ruolo collegato ai servizi per conto dell'utente.

Modificare un ruolo collegato al servizio per Amazon MSK

Amazon MSK non consente di modificare il ruolo collegato al servizio AWSServiceRoleForKafka. Dopo aver creato un ruolo collegato al servizio, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta la sezione Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi di Amazon MSK

Amazon MSK supporta l'utilizzo di ruoli collegati ai servizi in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta AWS Regioni ed endpoint.

AWS politiche gestite per Amazon MSK

Una politica AWS gestita è una politica autonoma creata e amministrata da. AWS AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni, in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Ti consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente specifiche per i tuoi casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove operazioni API per i servizi esistenti.

Per ulteriori informazioni, consultare Policy gestite da AWSnella Guida per l'utente di IAM.

AWS politica gestita: Amazon MSKFull Access

Questa policy concede autorizzazioni amministrative che consentono a un principale l'accesso completo a tutte le operazioni di Amazon MSK. Le autorizzazioni in questa policy sono raggruppate come segue:

- Le autorizzazioni Amazon MSK consentono tutte le operazioni di Amazon MSK.
- **`Amazon EC2`**autorizzazioni: in questa politica sono necessarie per convalidare le risorse passate in una richiesta API. Questo serve a garantire che Amazon MSK sia in grado di utilizzare correttamente le risorse di un cluster. Le altre EC2 autorizzazioni Amazon incluse in questa politica consentono ad Amazon MSK di creare AWS le risorse necessarie per consentirti di connetterti ai tuoi cluster.
- **`AWS KMS`**autorizzazioni: vengono utilizzate durante le chiamate API per convalidare le risorse passate in una richiesta. Sono necessarie per consentire ad Amazon MSK di utilizzare la chiave passata con il cluster Amazon MSK.

- **`CloudWatch Logs, Amazon S3, and Amazon Data Firehose`**autorizzazioni: sono necessarie per consentire ad Amazon MSK di garantire che le destinazioni di consegna dei log siano raggiungibili e che siano valide per l'utilizzo dei log da parte dei broker.
- **`IAM`**autorizzazioni: sono necessarie per consentire ad Amazon MSK di creare un ruolo collegato al servizio nel tuo account e per consentirti di passare un ruolo di esecuzione del servizio ad Amazon MSK.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [{
   "Effect": "Allow",
   "Action": [
    "kafka:*",
    "ec2:DescribeSubnets",
    "ec2:DescribeVpcs",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeRouteTables",
    "ec2:DescribeVpcEndpoints",
    "ec2:DescribeVpcAttribute",
    "kms:DescribeKey",
    "kms:CreateGrant",
    "logs:CreateLogDelivery",
    "logs:GetLogDelivery",
    "logs:UpdateLogDelivery",
    "logs:DeleteLogDelivery",
    "logs:ListLogDeliveries",
    "logs:PutResourcePolicy",
    "logs:DescribeResourcePolicies",
    "logs:DescribeLogGroups",
    "S3:GetBucketPolicy",
    "firehose:TagDeliveryStream"
   ],
   "Resource": "*"
  },
  {
   "Effect": "Allow",
   "Action": [
    "ec2:CreateVpcEndpoint"
   ],
```

```
        "Resource": [
         "arn:*:ec2:*:*:vpc/*",
         "arn:*:ec2:*:*:subnet/*",
         "arn:*:ec2:*:*:security-group/*"
        ]
       },
       {
        "Effect": "Allow",
        "Action": [
         "ec2:CreateVpcEndpoint"
        ],
        "Resource": [
         "arn:*:ec2:*:*:vpc-endpoint/*"
        ],
        "Condition": {
         "StringEquals": {
          "aws:RequestTag/AWSMSKManaged": "true"
         },
         "StringLike": {
          "aws:RequestTag/ClusterArn": "*"
         }
        }
       },
       {
        "Effect": "Allow",
        "Action": [
         "ec2:CreateTags"
        ],
        "Resource": "arn:*:ec2:*:*:vpc-endpoint/*",
        "Condition": {
         "StringEquals": {
          "ec2:CreateAction": "CreateVpcEndpoint"
         }
        }
       },
       {
        "Effect": "Allow",
        "Action": [
         "ec2:DeleteVpcEndpoints"
        ],
        "Resource": "arn:*:ec2:*:*:vpc-endpoint/*",
        "Condition": {
         "StringEquals": {
          "ec2:ResourceTag/AWSMSKManaged": "true"
```

```
        },
        "StringLike": {
         "ec2:ResourceTag/ClusterArn": "*"
        }
       }
      },
      {
       "Effect": "Allow",
       "Action": "iam:PassRole",
       "Resource": "*",
       "Condition": {
        "StringEquals": {
         "iam:PassedToService": "kafka.amazonaws.com"
        }
       }
      },
      {
       "Effect": "Allow",
       "Action": "iam:CreateServiceLinkedRole",
       "Resource": "arn:aws:iam::*:role/aws-service-role/kafka.amazonaws.com/
AWSServiceRoleForKafka*",
       "Condition": {
        "StringLike": {
         "iam:AWSServiceName": "kafka.amazonaws.com"
        }
       }
      },
      {
       "Effect": "Allow",
       "Action": [
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy"
       ],
       "Resource": "arn:aws:iam::*:role/aws-service-role/kafka.amazonaws.com/
AWSServiceRoleForKafka*"
      },
      {
       "Effect": "Allow",
       "Action": "iam:CreateServiceLinkedRole",
       "Resource": "arn:aws:iam::*:role/aws-service-role/
delivery.logs.amazonaws.com/AWSServiceRoleForLogDelivery*",
       "Condition": {
        "StringLike": {
         "iam:AWSServiceName": "delivery.logs.amazonaws.com"
```

```
        }
       }
      }

     ]
    }
```

AWS politica gestita: Amazon MSKRead OnlyAccess

Questa policy concede autorizzazioni di sola lettura che consentono agli utenti di visualizzare informazioni in Amazon MSK. I principali ai quali è collegata questa policy non possono effettuare aggiornamenti o eliminare risorse esistenti, né possono creare nuove risorse Amazon MSK. Ad esempio, i principali con queste autorizzazioni possono visualizzare l'elenco dei cluster e delle configurazioni associati al proprio account, ma non possono modificare la configurazione o le impostazioni di alcun cluster. Le autorizzazioni in questa policy sono raggruppate come segue:

- **`Amazon MSK`**autorizzazioni: consentono di elencare le risorse Amazon MSK, descriverle e ottenere informazioni su di esse.
- **`Amazon EC2`**autorizzazioni: vengono utilizzate per descrivere Amazon VPC, le sottoreti, i gruppi di sicurezza ENIs e sono associati a un cluster.
- **`AWS KMS`**autorizzazione: viene utilizzata per descrivere la chiave associata al cluster.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "kafka:Describe*",
                "kafka:List*",
                "kafka:Get*",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "kms:DescribeKey"
            ],
            "Effect": "Allow",
```

```
            "Resource": "*"
        }
    ]
}
```

AWS politica gestita: KafkaServiceRolePolicy

Non puoi collegarti KafkaServiceRolePolicy alle tue entità IAM. Questa policy è collegata a un ruolo collegato ai servizi che consente ad Amazon MSK di eseguire operazioni come la gestione degli endpoint VPC (connettori) nei cluster MSK, la gestione delle interfacce di rete e la gestione delle credenziali del cluster con AWS Secrets Manager. Per ulteriori informazioni, consulta the section called “Ruoli collegati ai servizi”.

AWS politica gestita: AWSMSKReplicator ExecutionRole

La `AWSMSKReplicatorExecutionRole` policy concede le autorizzazioni al replicatore Amazon MSK per replicare i dati tra cluster MSK. Le autorizzazioni in questa policy sono raggruppate come segue:

- **`cluster`**— Concede ad Amazon MSK Replicator le autorizzazioni per connettersi al cluster utilizzando l'autenticazione IAM. Concede inoltre le autorizzazioni per descrivere e modificare il cluster.
- **`topic`**— Concede ad Amazon MSK Replicator le autorizzazioni per descrivere, creare e modificare un argomento e per modificare la configurazione dinamica dell'argomento.
- **`consumer group`**— Concede ad Amazon MSK Replicator le autorizzazioni per descrivere e modificare gruppi di consumatori, leggere e scrivere dati da un cluster MSK e eliminare argomenti interni creati dal replicatore.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "ClusterPermissions",
   "Effect": "Allow",
   "Action": [
    "kafka-cluster:Connect",
```

```
    "kafka-cluster:DescribeCluster",
    "kafka-cluster:AlterCluster",
    "kafka-cluster:DescribeTopic",
    "kafka-cluster:CreateTopic",
    "kafka-cluster:AlterTopic",
    "kafka-cluster:WriteData",
    "kafka-cluster:ReadData",
    "kafka-cluster:AlterGroup",
    "kafka-cluster:DescribeGroup",
    "kafka-cluster:DescribeTopicDynamicConfiguration",
    "kafka-cluster:AlterTopicDynamicConfiguration",
    "kafka-cluster:WriteDataIdempotently"
   ],
   "Resource": [
    "arn:aws:kafka:*:*:cluster/*"
   ]
  },
  {
   "Sid": "TopicPermissions",
   "Effect": "Allow",
   "Action": [
    "kafka-cluster:DescribeTopic",
    "kafka-cluster:CreateTopic",
    "kafka-cluster:AlterTopic",
    "kafka-cluster:WriteData",
    "kafka-cluster:ReadData",
    "kafka-cluster:DescribeTopicDynamicConfiguration",
    "kafka-cluster:AlterTopicDynamicConfiguration",
    "kafka-cluster:AlterCluster"
   ],
   "Resource": [
    "arn:aws:kafka:*:*:topic/*/*"
   ]
  },
  {
   "Sid": "GroupPermissions",
   "Effect": "Allow",
   "Action": [
    "kafka-cluster:AlterGroup",
    "kafka-cluster:DescribeGroup"
   ],
   "Resource": [
    "arn:aws:kafka:*:*:group/*/*"
   ]
```

```
    }
  ]
}
```

Amazon MSK aggiorna le politiche AWS gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite per Amazon MSK da quando questo servizio ha iniziato a tracciare queste modifiche.

Modifica	Descrizione	Data
WriteDataIdempotently autorizzazione aggiunta a AWSMSKReplicator Execution Role: aggiornamento a una politica esistente	Amazon MSK ha aggiunto WriteDataIdempoten tly l'autorizzazione alla AWSMSKReplicator Execution Role policy per supportare la replica dei dati tra cluster MSK.	12 marzo 2024
AWSMSKReplicatorEx ecutionRole: nuova policy	Amazon MSK ha aggiunto una AWSMSKReplicator Execution Role policy per supportare Amazon MSK Replicator.	4 dicembre 2023
Amazon MSKFull Access: aggiornamento a una politica esistente	Amazon MSK ha aggiunto le autorizzazioni per supportare il replicatore Amazon MSK.	28 settembre 2023
KafkaServiceRolePolicy: aggiornamento a una policy esistente	Amazon MSK ha aggiunto le autorizzazioni per supportar e la connettività privata multi-VPC.	8 marzo 2023
Amazon MSKFull Access: aggiornamento a una politica esistente	Amazon MSK ha aggiunto nuove EC2 autorizzazioni Amazon per consentire la connessione a un cluster.	30 novembre 2021

Modifica	Descrizione	Data
Amazon MSKFull Access: aggiornamento a una politica esistente	Amazon MSK ha aggiunto una nuova autorizzazione per consentirgli di descriver e le tabelle di EC2 routing di Amazon.	19 novembre 2021
Amazon MSK ha iniziato a tenere traccia delle modifiche	Amazon MSK ha iniziato a tracciare le modifiche per le sue politiche AWS gestite.	19 novembre 2021

Risolvi i problemi relativi all'identità e all'accesso ad Amazon MSK

Utilizza le informazioni seguenti per eseguire la diagnosi e risolvere i problemi comuni che possono verificarsi durante l'utilizzo di Amazon MSK e IAM.

Argomenti

- Non dispongo dell'autorizzazione per eseguire un'operazione in Amazon MSK

Non dispongo dell'autorizzazione per eseguire un'operazione in Amazon MSK

Se ti AWS Management Console dice che non sei autorizzato a eseguire un'azione, devi contattare l'amministratore per ricevere assistenza. L'amministratore è colui che ti ha fornito le credenziali di accesso.

L'errore di esempio seguente si verifica quando l'utente IAM `mateojackson` cerca di utilizzare la console per eliminare un cluster senza disporre delle autorizzazioni `kafka:DeleteCluster`.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 kafka:DeleteCluster on resource: purchaseQueriesCluster
```

In questo caso, Mateo richiede al suo amministratore di aggiornare le policy per poter accedere alla risorsa `purchaseQueriesCluster` utilizzando l'azione `kafka:DeleteCluster`.

Autenticazione e autorizzazione per Apache Kafka APIs

Puoi utilizzare IAM per autenticare i client e consentire o rifiutare le operazioni di Apache Kafka. In alternativa, puoi utilizzare TLS SASL/SCRAM per autenticare i client e Apache Kafka per consentire o negare azioni. ACLs

Per informazioni su come controllare chi può eseguire le operazioni di Amazon MSK sul tuo cluster, consulta la pagina the section called “Autenticazione e autorizzazione per Amazon MSK APIs”.

Argomenti

- Controllo degli accessi IAM
- Autenticazione client TLS reciproca per Amazon MSK
- Autenticazione delle credenziali di accesso con Secrets Manager AWS
- Apache Kafka ACLs

Controllo degli accessi IAM

Il Controllo degli accessi IAM per Amazon MSK ti consente di gestire sia l'autenticazione sia l'autorizzazione per il cluster MSK. Ciò elimina la necessità di utilizzare meccanismi separati per l'autenticazione e l'autorizzazione. Ad esempio, quando un client tenta di scrivere sul cluster, Amazon MSK utilizza IAM per verificare se tale client è un'identità autenticata e se è autorizzato a produrre nel cluster.

Il controllo degli accessi IAM funziona per client Java e non Java, inclusi i client Kafka scritti in Python, Go e.NET. JavaScript Il controllo degli accessi IAM per client non Java è disponibile per i cluster MSK con Kafka versione 2.7.1 o successiva.

Per rendere possibile il Controllo degli accessi IAM, Amazon MSK apporta piccole modifiche al codice sorgente di Apache Kafka. Queste modifiche non causeranno differenze evidenti nella tua esperienza con Apache Kafka. Amazon MSK registra gli eventi di accesso in modo da poterli controllare.

È possibile richiamare Apache Kafka ACL per un cluster MSK che utilizza il controllo degli accessi IAM. APIs Tuttavia, Apache Kafka ACLs non ha alcun effetto sull'autorizzazione per le identità IAM. È necessario utilizzare le policy IAM per controllare l'accesso alle identità IAM.

⚠ Considerazioni importanti

Quando utilizzi il controllo degli accessi IAM con il tuo cluster MSK, tieni presenti le seguenti importanti considerazioni:

- Il controllo degli accessi IAM non si applica ai nodi ZooKeeper Apache. Per ulteriori informazioni sul controllo degli accessi a tali nodi, consulta la pagina Controlla l'accesso ai ZooKeeper nodi Apache nel tuo cluster Amazon MSK.
- L'impostazione `allow.everyone.if.no.acl.found` di Apache Kafka non ha effetto se il cluster utilizza il Controllo degli accessi IAM.
- Puoi richiamare Apache Kafka ACL APIs per un cluster MSK che utilizza il controllo degli accessi IAM. Tuttavia, Apache Kafka ACLs non ha alcun effetto sull'autorizzazione per le identità IAM. È necessario utilizzare le policy IAM per controllare l'accesso alle identità IAM.

Come funziona il Controllo degli accessi IAM per Amazon MSK

Per utilizzare il controllo degli accessi IAM per Amazon MSK, esegui i seguenti passaggi, descritti in dettaglio in questi argomenti:

- Crea un cluster Amazon MSK che utilizza il controllo degli accessi IAM
- Configurazione dei client per il Controllo degli accessi IAM
- Crea politiche di autorizzazione per il ruolo IAM
- Recupero dei broker di bootstrap per il Controllo degli accessi IAM

Crea un cluster Amazon MSK che utilizza il controllo degli accessi IAM

Questa sezione spiega come utilizzare AWS Management Console, l'API o il AWS CLI per creare un cluster Amazon MSK che utilizza il controllo degli accessi IAM. Per informazioni su come attivare il Controllo degli accessi IAM per un cluster esistente, consulta la pagina Aggiornamento delle impostazioni di sicurezza di un cluster Amazon MSK.

Utilizza il AWS Management Console per creare un cluster che utilizza il controllo degli accessi IAM

1. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/.
2. Scegli Create cluster (Crea cluster).
3. Scegli Crea cluster con impostazioni personalizzate.
4. Nella sezione Autenticazione, scegli Controllo degli accessi IAM.
5. Completa il resto del flusso di lavoro per creare un cluster.

Utilizza l'API o il AWS CLI per creare un cluster che utilizza il controllo degli accessi IAM

- Per creare un cluster con il controllo degli accessi IAM abilitato, utilizza l'CreateClusterAPI o il comando CLI create-cluster e passa il seguente JSON per il parametro:. `ClientAuthentication` `"ClientAuthentication": { "Sasl": { "Iam": { "Enabled": true } }`

Configurazione dei client per il Controllo degli accessi IAM

Per consentire ai client di comunicare con un cluster MSK che utilizza il controllo degli accessi IAM, è possibile utilizzare uno di questi meccanismi:

- Configurazione client non Java tramite meccanismo SASL_OAUTHBEARER
- Configurazione del client Java mediante SASL_OAUTHBEARER meccanismo o AWS_MSK_IAM meccanismo

Usa il SASL_OAUTHBEARER meccanismo per configurare IAM

1. Modifica il tuo file di configurazione client.properties usando il seguente esempio di client Python Kafka. Le modifiche alla configurazione sono simili in altri linguaggi.

```
from kafka import KafkaProducer
from kafka.errors import KafkaError
from kafka.sasl.oauth import AbstractTokenProvider
import socket
import time
from aws_msk_iam_sasl_signer import MSKAuthTokenProvider

class MSKTokenProvider():
    def token(self):
        token, _ = MSKAuthTokenProvider.generate_auth_token('<my Regione AWS>')
        return token

tp = MSKTokenProvider()

producer = KafkaProducer(
    bootstrap_servers='<myBootstrapString>',
    security_protocol='SASL_SSL',
    sasl_mechanism='OAUTHBEARER',
    sasl_oauth_token_provider=tp,
```

```
    client_id=socket.gethostname(),
)

topic = "<my-topic>"
while True:
    try:
        inp=input(">")
        producer.send(topic, inp.encode())
        producer.flush()
        print("Produced!")
    except Exception:
        print("Failed to send message:", e)

producer.close()
```

2. Scarica la libreria di supporto per la lingua di configurazione scelta e segui le istruzioni nella sezione Guida introduttiva della home page della libreria di lingue.

   - JavaScript: https://github.com/aws/aws-msk-iam-sasl-signer-js #getting -started
   - Python: https://github.com/aws/aws-msk-iam-sasl-signer-python #get -avviato
   - Vai: https://github.com/aws/aws-msk-iam-sasl-signer-go #getting -started
   - .NET: -signer-net #getting -iniziato https://github.com/aws/ aws-msk-iam-sasl
   - JAVA: SASL_OAUTHBEARER il supporto per Java è disponibile tramite il file jar `aws-msk-iam-auth`

Utilizza il AWS_MSK_IAM meccanismo personalizzato MSK per configurare IAM

1. Aggiungi quanto segue al file `client.properties`. Sostituisci *<PATH_TO_TRUST_STORE_FILE>* con il percorso completo del file di trust store sul client.

   > **Note**
   >
   > Se non desideri utilizzare un certificato specifico, puoi rimuovere `ssl.truststore.location=`*<PATH_TO_TRUST_STORE_FILE>* dal tuo file `client.properties`. Se non specifichi un valore per `ssl.truststore.location`, il processo Java utilizza il certificato predefinito.

```
ssl.truststore.location=<PATH_TO_TRUST_STORE_FILE>
```

```
security.protocol=SASL_SSL
sasl.mechanism=AWS_MSK_IAM
sasl.jaas.config=software.amazon.msk.auth.iam.IAMLoginModule required;
sasl.client.callback.handler.class=software.amazon.msk.auth.iam.IAMClientCallbackHandler
```

Per utilizzare un profilo denominato creato per AWS le credenziali, includilo `awsProfileName="your profile name";` nel file di configurazione del client. Per informazioni sui profili denominati, consulta Profili denominati nella AWS CLI documentazione.

2. Scaricate l'ultimo file aws-msk-iam-authJAR stabile e inseritelo nel percorso della classe. Se utilizzi Maven, aggiungi la seguente dipendenza, modificando il numero di versione secondo necessità:

```
<dependency>
    <groupId>software.amazon.msk</groupId>
    <artifactId>aws-msk-iam-auth</artifactId>
    <version>1.0.0</version>
</dependency>
```

Il plug-in client di Amazon MSK è open source con licenza Apache 2.0.

Crea politiche di autorizzazione per il ruolo IAM

Collega una policy di autorizzazione al ruolo IAM corrispondente al client. In una policy di autorizzazione, specifichi quali operazioni consentire o rifiutare per il ruolo. Se il tuo cliente utilizza un' EC2 istanza Amazon, associa la politica di autorizzazione al ruolo IAM per quell' EC2 istanza Amazon. In alternativa, puoi configurare il client per utilizzare un profilo denominato e quindi associare la policy di autorizzazione al ruolo per quel profilo denominato. Configurazione dei client per il Controllo degli accessi IAM descrive come configurare un client per utilizzare un profilo denominato.

Per informazioni sulla creazione di una policy IAM, consulta la pagina Creating IAM policies.

Di seguito è riportato un esempio di politica di autorizzazione per un cluster denominato MyTestCluster. Per comprendere la semantica degli elementi `Action` e `Resource`, consulta la pagina Semantica delle azioni e delle risorse della politica di autorizzazione IAM.

> **Important**
>
> Le modifiche apportate a una policy IAM si riflettono nell'IAM APIs e nell' AWS CLI immediatezza. Tuttavia, può trascorrere molto tempo prima che la modifica della policy abbia effetto. Nella maggior parte dei casi, le modifiche alle policy entrano in vigore in meno di un minuto. A volte le condizioni della rete possono aumentare il ritardo.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:AlterCluster",
                "kafka-cluster:DescribeCluster"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:111122223333:cluster/MyTestCluster/
abcd1234-0123-abcd-5678-1234abcd-1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:*Topic*",
                "kafka-cluster:WriteData",
                "kafka-cluster:ReadData"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:123456789012:topic/MyTestCluster/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeGroup"
```

```
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:123456789012:group/MyTestCluster/*"
            ]
        }
    ]
}
```

Per informazioni su come creare una policy con elementi di operazione che corrispondano ai casi d'uso comuni di Apache Kafka, come la produzione e l'utilizzo di dati, consulta la pagina Casi d'uso comuni per la politica di autorizzazione dei client.

Per le versioni di Kafka 2.8.0 e successive, l'WriteDataIdempotentlyautorizzazione è obsoleta (KIP-679). Per impostazione predefinita, viene utilizzato `enable.idempotence = true`. Pertanto, per le versioni di Kafka 2.8.0 e successive, IAM non offre le stesse funzionalità di Kafka. ACLs Non è possibile accedere a un argomento fornendo solo `WriteDataIdempotently` l'accesso a quell'argomento. `WriteData` Ciò non influisce sul caso in cui `WriteData` venga fornito a TUTTI gli argomenti. In tal caso, l'operazione `WriteDataIdempotently` è consentita. Ciò è dovuto alle differenze nell'implementazione della logica IAM e nel modo in cui ACLs vengono implementati i Kafka. Inoltre, scrivere su un argomento in modo idempotente richiede anche l'accesso a. `transactional-ids`

Per ovviare a questo problema, consigliamo di utilizzare una politica simile alla seguente politica.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:AlterCluster",
                "kafka-cluster:DescribeCluster",
                "kafka-cluster:WriteDataIdempotently"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:123456789012:cluster/MyTestCluster/
abcd1234-0123-abcd-5678-1234abcd-1"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:*Topic*",
                "kafka-cluster:WriteData",
                "kafka-cluster:ReadData"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:123456789012:topic/MyTestCluster/
abcd1234-0123-abcd-5678-1234abcd-1/TestTopic",
                "arn:aws:kafka:us-east-1:123456789012:transactional-id/
MyTestCluster/abcd1234-0123-abcd-5678-1234abcd-1/*"
            ]
        }
    ]
}
```

In questo caso, `WriteData` consente le scritture sul `TestTopic`, mentre `WriteDataIdempotently` consente le scritture idempotenti sul cluster. Questa politica aggiunge anche l'accesso alle `transactional-id` risorse che saranno necessarie.

Poiché `WriteDataIdempotently` si tratta di un'autorizzazione a livello di cluster, non è possibile utilizzarla a livello di argomento. Se `WriteDataIdempotently` è limitato al livello di argomento, questo criterio non funzionerà.

Recupero dei broker di bootstrap per il Controllo degli accessi IAM

Consultare Ottieni i broker bootstrap per un cluster Amazon MSK.

Semantica delle azioni e delle risorse della politica di autorizzazione IAM

Attualmente, il controllo degli accessi IAM per Amazon MSK non supporta le azioni di cluster interne per Kafka. Ciò include l' WriteTxnMarkers API, che Kafka utilizza per terminare le transazioni. Per terminare le transazioni, ti consigliamo di utilizzare l'autenticazione SCRAM o MTLS con l'autenticazione appropriata anziché l'autenticazione IAM. ACLs

In questa sezione viene illustrata la semantica degli elementi di operazione e risorsa che è possibile utilizzare in una policy di autorizzazione IAM. Per un esempio di policy, consulta Crea politiche di autorizzazione per il ruolo IAM.

Azioni relative alla politica di autorizzazione

La tabella seguente elenca le operazioni che è possibile includere in una policy di autorizzazione quando si utilizza il Controllo degli accessi IAM per Amazon MSK. Quando includi nella tua policy di autorizzazione un'operazione dalla colonna Operazione della tabella, devi includere anche le operazioni corrispondenti dalla colonna Operazioni richieste.

Azione	Descrizione	Operazioni necessarie	Risorse obbligatorie	Applicabile ai cluster serverless
`kafka-cluster:Connect`	Concede l'autorizzazione per connettersi e autenticarsi al cluster.	Nessuno	cluster	Sì
`kafka-cluster:DescribeCluster`	Concede l'autorizzazione per descrivere vari aspetti del cluster, equivalente all'ACL DESCRIBE CLUSTER di Apache Kafka.	`kafka-cluster:Connect`	cluster	Sì
`kafka-cluster:AlterCluster`	Concede l'autorizzazione per modificare vari aspetti del cluster, equivalente all'ACL ALTER CLUSTER di Apache Kafka.	`kafka-cluster:Connect` `kafka-cluster:DescribeCluster`	cluster	No

Azione	Descrizione	Operazioni necessarie	Risorse obbligatorie	Applicabile ai cluster serverless
`kafka-cluster:DescribeClusterDynamicConfiguration`	Concede l'autorizzazione per descrivere la configurazione dinamica di un cluster, equivalente all'ACL DESCRIBE_CONFIGS CLUSTER di Apache Kafka.	`kafka-cluster:Connect`	cluster	No

`kafka-cluster:AlterClusterDynamicConfiguration`	Concede l'autorizzazione per modificare la configurazione dinamica di un cluster, equivalente all'ACL ALTER_CONFIGS CLUSTER di Apache Kafka.	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeClusterDynamicConfiguration`
			cluster	No

Azione	Descrizione	Operazioni necessarie	Risorse obbligatorie	Applicabile ai cluster serverless
`kafka-cluster:WriteDataIdempotently`	Concede l'autorizzazione per scrivere dati in modo idempotente su un cluster, equivalente all'ACL IDEMPOTENT_WRITE CLUSTER di Apache Kafka.	`kafka-cluster:Connect` `kafka-cluster:WriteData`	cluster	Sì
`kafka-cluster:CreateTopic`	Concede l'autorizzazione a creare argomenti su un cluster, equivalente all'ACL CREATE di Apache Kafka. CLUSTER/TOPIC	`kafka-cluster:Connect`	topic	Sì
`kafka-cluster:DescribeTopic`	Concede l'autorizzazione per descrivere gli argomenti in un cluster, equivalente all'ACL DESCRIBE TOPIC di Apache Kafka.	`kafka-cluster:Connect`	topic	Sì

Azione	Descrizione	Operazioni necessarie	Risorse obbligatorie	Applicabile ai cluster serverless
`kafka-cluster:AlterTopic`	Concede l'autorizzazione per modificare gli argomenti in un cluster, equivalente all'ACL ALTER TOPIC di Apache Kafka.	`kafka-cluster:Connect` `kafka-cluster:DescribeTopic`	topic	Sì
`kafka-cluster:DeleteTopic`	Concede l'autorizzazione per eliminare gli argomenti in un cluster, equivalente all'ACL DELETE TOPIC di Apache Kafka.	`kafka-cluster:Connect` `kafka-cluster:DescribeTopic`	topic	Sì
`kafka-cluster:DescribeTopicDynamicConfiguration`	Concede l'autorizzazione per descrivere la configurazione dinamica degli argomenti in un cluster, equivalente all'ACL DESCRIBE_CONFIGS TOPIC di Apache Kafka.	`kafka-cluster:Connect`	topic	Sì

Azione	Descrizione	Operazioni necessarie	Risorse obbligatorie	Applicabile ai cluster serverless
`kafka-cluster:AlterTopicDynamicConfiguration`	Concede l'autorizzazione per modificare la configurazione dinamica degli argomenti in un cluster, equivalente all'ACL ALTER_CONFIGS TOPIC di Apache Kafka.	`kafka-cluster:Connect` `kafka-cluster:DescribeTopicDynamicConfiguration`	topic	Sì

`kafka-cluster:ReadData`	Concede l'autorizzazione per leggere i dati da argomenti in un cluster, equivalente all'ACL READ TOPIC di Apache Kafka.	`kafka-cluster:Connect`
		`kafka-cluster:DescribeTopic`
		`kafka-cluster:AlterGroup`
			topic	Sì

Azione	Descrizione	Operazioni necessarie	Risorse obbligatorie	Applicabile ai cluster serverless

`kafka-cluster:WriteData`	Concede l'autorizzazione per scrivere dati su argomenti in un cluster, equivalente all'ACL WRITE TOPIC di Apache Kafka	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeTopic`
			topic	Sì

`kafka-cluster:DescribeGroup`	Concede l'autorizzazione per descrivere i gruppi in un cluster, equivalente all'ACL DESCRIBE GROUP di Apache Kafka.	`kafka-cluster:Connect`	gruppo	Sì

`kafka-cluster:AlterGroup`	Concede l'autorizzazione per unire dei gruppi all'interno di un cluster, equivalente all'ACL READ GROUP di Apache Kafka.	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeGroup`
			gruppo	Sì

Azione	Descrizione	Operazioni necessarie	Risorse obbligatorie	Applicabile ai cluster serverless

`kafka-cluster:DeleteGroup`	Concede l'autorizzazione per eliminare gruppi all'interno di un cluster, equivalente all'ACL DELETE GROUP di Apache Kafka.	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeGroup`
			gruppo	Sì

`kafka-cluster:DescribeTransactionalId`	Concede l'autorizzazione a descrivere le transazioni IDs su un cluster, equivalente all'ACL DESCRIBE TRANSACTIONAL_ID di Apache Kafka.	`kafka-cluster:Connect`	transactional-id	Sì

Azione	Descrizione	Operazioni necessarie	Risorse obbligatorie	Applicabile ai cluster serverless

`kafka-cluster:AlterTransactionalId`	Concede l'autorizzazione a modificare le transazioni su un cluster, equivalente all'ACL WRITE IDs TRANSACTIONAL_ID di Apache Kafka.	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeTransactionalId`
		
		`kafka-cluster:WriteData`
			transactional-id	Sì

In un'operazione, dopo i due punti, è possibile utilizzare qualsiasi quantità di caratteri jolly asterisco (*). Di seguito vengono mostrati gli esempi.

- `kafka-cluster:*Topic` sta per `kafka-cluster:CreateTopic`, `kafka-cluster:DescribeTopic`, `kafka-cluster:AlterTopic` e `kafka-cluster:DeleteTopic`. Non include `kafka-cluster:DescribeTopicDynamicConfiguration` o `kafka-cluster:AlterTopicDynamicConfiguration`.
- `kafka-cluster:*` indica tutte le autorizzazioni.

Risorse relative alla politica di autorizzazione

La tabella seguente mostra i quattro tipi di risorse che è possibile utilizzare in una policy di autorizzazione quando si utilizza il Controllo degli accessi IAM per Amazon MSK. Puoi ottenere il cluster Amazon Resource Name (ARN) da o utilizzando l'DescribeClusterAPI AWS Management Console o il comando AWS CLI describe-cluster. È quindi possibile utilizzare l'ARN del cluster per creare un argomento, un gruppo e un ID transazionale. ARNs Per specificare una risorsa nella policy di autorizzazione, utilizza l'ARN della risorsa.

Risorsa	Formato ARN
Cluster	arn:aws:kafka: :cluster//*regionaccount-id cluster-name cluster-uuid*
Argomento	arn:aws:kafka: *region* ::argomento///*account-id cluster-name cluster-uuid topic-name*
Group (Gruppo)	arn:aws:kafka: *region* :group///*account-id cluster-name cluster-uuid group-name*
ID transazionale	arn:aws:kafka: *region* ::identificativo-transazione///*account-id cluster-name cluster-uuid transactional-id*

È possibile utilizzare qualsiasi quantità di caratteri jolly asterisco (*) in qualsiasi punto dell'ARN successivo a `:cluster/`, `:topic/`, `:group/` e `:transactional-id/`. Di seguito sono riportati alcuni esempi di come utilizzare il carattere jolly asterisco (*) per fare riferimento a più risorse:

- `arn:aws:kafka:us-east-1:0123456789012:topic/MyTestCluster/*`: tutti gli argomenti di qualsiasi cluster denominato, indipendentemente dall'UUID del cluster. MyTestCluster
- `arn:aws:kafka:us-east-1:0123456789012:topic/MyTestCluster/abcd1234-0123-abcd-5678-1234abcd-1/*_test`: tutti gli argomenti il cui nome termina con «_test» nel cluster il cui nome è MyTestCluster e il cui UUID è abcd1234-0123-abcd-5678-1234abcd-1.
- `arn:aws:kafka:us-east-1:0123456789012:transactional-id/MyTestCluster/*/5555abcd-1111-abcd-1234-abcd1234-1`: tutte le transazioni il cui ID transazionale è MyTestCluster 5555abcd-1111-abcd-1234-abcd1234-1, in tutte le incarnazioni di un cluster denominato nel tuo account. Ciò significa che se si crea un cluster denominato MyTestCluster, quindi lo si elimina e quindi si crea un altro cluster con lo stesso nome, è possibile utilizzare questa risorsa ARN per rappresentare lo stesso ID delle transazioni su entrambi i cluster. Tuttavia, il cluster eliminato non è accessibile.

Casi d'uso comuni per la politica di autorizzazione dei client

La prima colonna della tabella seguente mostra alcuni casi d'uso comuni. Per autorizzare un client a eseguire un determinato caso d'uso, includi le operazioni richieste per tale caso d'uso nella policy di autorizzazione del client e imposta `Effect` su `Allow`.

Per informazioni su tutte le operazioni che fanno parte del Controllo degli accessi IAM per Amazon MSK, consulta la pagina Semantica delle azioni e delle risorse della politica di autorizzazione IAM.

> **Note**
>
> Le operazioni non sono consentite per impostazione predefinita. È necessario consentire esplicitamente ogni operazione che si desidera autorizzare il client a eseguire.

Caso d'uso	Operazioni necessarie
Admin	`kafka-cluster:*`
Creazione di un argomento	`kafka-cluster:Connect` `kafka-cluster:CreateTopic`

Produzione di dati	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:WriteData`

Utilizzo di dati	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:DescribeGroup`
	`kafka-cluster:AlterGroup`
	`kafka-cluster:ReadData`

Produzione di dati in modo idempotente	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:WriteData`
	`kafka-cluster:WriteDataIdempotently`

Caso d'uso	Operazioni necessarie

Produzione di dati in modo transazionale	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:WriteData`
	`kafka-cluster:DescribeTransactionalId`
	`kafka-cluster:AlterTransactionalId`

Descrizione della configurazione di un cluster	`kafka-cluster:Connect` `kafka-cluster:DescribeClusterDynamicConfiguration`

Aggiornamento della configurazione di un cluster	`kafka-cluster:Connect`
	`kafka-cluster:DescribeClusterDynamicConfiguration`
	`kafka-cluster:AlterClusterDynamicConfiguration`

Descrizione della configurazione di un argomento	`kafka-cluster:Connect` `kafka-cluster:DescribeTopicDynamicConfiguration`

Aggiornamento della configurazione di un argomento	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopicDynamicConfiguration`
	`kafka-cluster:AlterTopicDynamicConfiguration`

Caso d'uso	Operazioni necessarie

Modifica di un argomento	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:AlterTopic`

Autenticazione client TLS reciproca per Amazon MSK

Puoi abilitare l'autenticazione client con TLS per le connessioni dalle tue applicazioni ai broker Amazon MSK. Per utilizzare l'autenticazione client, è necessario un CA privata AWS. CA privata AWS Possono appartenere allo Account AWS stesso cluster o a un account diverso. Per informazioni su CA privata AWS s, vedere Creazione e gestione di un CA privata AWS.

> **Note**
>
> L'autenticazione TLS non è disponibile nelle regioni di Pechino e Ningxia.

Amazon MSK non supporta gli elenchi di revoca dei certificati ()CRLs. Per controllare l'accesso agli argomenti del cluster o bloccare i certificati compromessi, usa Apache ACLs Kafka e i gruppi di sicurezza. AWS Per informazioni sull'uso di Apache Kafka, consulta. ACLs the section called “Apache Kafka ACLs”

Questo argomento contiene le sezioni seguenti:

- Crea un cluster Amazon MSK che supporti l'autenticazione dei client
- Configura un client per utilizzare l'autenticazione
- Produci e consuma messaggi utilizzando l'autenticazione

Crea un cluster Amazon MSK che supporti l'autenticazione dei client

Questa procedura mostra come abilitare l'autenticazione del client utilizzando un CA privata AWS.

> **Note**
>
> Si consiglia vivamente di utilizzare Independent CA privata AWS per ogni cluster MSK quando si utilizza il TLS reciproco per controllare l'accesso. In questo modo assicurerai che i certificati TLS firmati da si autentichino PCAs solo con un singolo cluster MSK.

1. Crea un file denominato `clientauthinfo.json` con i seguenti contenuti. Sostituisci *`Private-CA-ARN`* con l'ARN del tuo PCA.

```
{
   "Tls": {
       "CertificateAuthorityArnList": ["Private-CA-ARN"]
    }
}
```

2. Crea un file denominato `brokernodegroupinfo.json` come descritto in the section called “Crea un cluster Amazon MSK con provisioning utilizzando AWS CLI”.
3. L'autenticazione client richiede di abilitare anche la crittografia dei dati in transito tra client e broker. Crea un file denominato `encryptioninfo.json` con i seguenti contenuti. Sostituisci *`KMS-Key-ARN`* con l'ARN della tua chiave KMS. Puoi impostare `ClientBroker` su `TLS` o `TLS_PLAINTEXT`.

```
{
   "EncryptionAtRest": {
       "DataVolumeKMSKeyId": "KMS-Key-ARN"
    },
   "EncryptionInTransit": {
        "InCluster": true,
        "ClientBroker": "TLS"
    }
}
```

   Per ulteriori informazioni sulla crittografia, consulta the section called “Crittografia di Amazon MSK”.
4. Su una macchina su cui è AWS CLI installata, esegui il comando seguente per creare un cluster con l'autenticazione e la crittografia in transito abilitate. Salva l'ARN del cluster fornito nella risposta.

```
aws kafka create-cluster --cluster-name "AuthenticationTest" --broker-node-group-
info file://brokernodegroupinfo.json --encryption-info file://encryptioninfo.json
 --client-authentication file://clientauthinfo.json --kafka-version "{YOUR KAFKA
 VERSION}" --number-of-broker-nodes 3
```

Configura un client per utilizzare l'autenticazione

Questo processo descrive come configurare un' EC2 istanza Amazon da utilizzare come client per utilizzare l'autenticazione.

Questo processo descrive come produrre e utilizzare messaggi utilizzando l'autenticazione creando una macchina client, creando un argomento e configurando le impostazioni di sicurezza richieste.

1. Crea un' EC2 istanza Amazon da utilizzare come macchina client. Per semplicità, creare questa istanza nello stesso VPC utilizzato per il cluster. Consulta the section called “Crea una macchina client” per un esempio di come creare un computer client di questo tipo.
2. Creazione di un argomento. Per un esempio, consulta le istruzioni in the section called “Creazione di un argomento”.
3. Su una macchina su cui l'hai AWS CLI installato, esegui il seguente comando per ottenere i broker bootstrap del cluster. Sostituisci *Cluster-ARN* con l'ARN del tuo cluster.

   ```
   aws kafka get-bootstrap-brokers --cluster-arn Cluster-ARN
   ```

   Salvare la stringa associata a `BootstrapBrokerStringTls` nella risposta.
4. Sul computer client, eseguire il comando seguente per utilizzare il truststore JVM per creare il truststore client. Se il percorso JVM è diverso, modificare il comando di conseguenza.

   ```
   cp /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.201.b09-0.amzn2.x86_64/jre/lib/security/
   cacerts kafka.client.truststore.jks
   ```
5. Sul computer client, eseguire il comando seguente per creare una chiave privata per il client. Sostituisci *Distinguished-NameExample-Alias*,*Your-Store-Pass*, e *Your-Key-Pass* con stringhe a tua scelta.

   ```
   keytool -genkey -keystore kafka.client.keystore.jks -validity 300 -storepass Your-
   Store-Pass -keypass Your-Key-Pass -dname "CN=Distinguished-Name" -alias Example-
   Alias -storetype pkcs12 -keyalg rsa
   ```

6. Sul computer client, eseguire il comando seguente per creare una richiesta di certificato con la chiave privata creata nella fase precedente.

```
keytool -keystore kafka.client.keystore.jks -certreq -file client-cert-sign-request
 -alias Example-Alias -storepass Your-Store-Pass -keypass Your-Key-Pass
```

7. Aprire il file `client-cert-sign-request` e accertarsi che inizi con `-----BEGIN CERTIFICATE REQUEST-----` e termini con `-----END CERTIFICATE REQUEST-----`. Se inizia con `-----BEGIN NEW CERTIFICATE REQUEST-----`, eliminare la parola `NEW` (e il singolo spazio che la segue) dall'inizio e dalla fine del file.

8. Su un computer in cui è AWS CLI installato, esegui il comando seguente per firmare la richiesta di certificato. Sostituisci *Private-CA-ARN* con l'ARN del tuo PCA. Se lo si desidera, è possibile modificare il valore di validità. In questo esempio viene utilizzato 300.

```
aws acm-pca issue-certificate --certificate-authority-arn Private-CA-ARN --csr
 fileb://client-cert-sign-request --signing-algorithm "SHA256WITHRSA" --validity
 Value=300,Type="DAYS"
```

   Salvare il certificato ARN fornito nella risposta.

> **Note**
>
> Per recuperare il certificato client, utilizza il comando `acm-pca get-certificate` e specifica l'ARN del certificato. Per ulteriori informazioni, consulta la sezione get-certificate nella documentazione di riferimento alla AWS CLI .

9. Esegui il comando seguente per ottenere il certificato CA privata AWS firmato per te. Sostituisci *Certificate-ARN* con l'ARN ottenuto dalla risposta al comando precedente.

```
aws acm-pca get-certificate --certificate-authority-arn Private-CA-ARN --
certificate-arn Certificate-ARN
```

10. Dal risultato JSON dell'esecuzione del comando precedente, copiare le stringhe associate a `Certificate` e `CertificateChain`. Incolla queste due stringhe in un nuovo file denominato. signed-certificate-from-acm Incollare innanzitutto la stringa associata a `Certificate`, seguita dalla stringa associata a `CertificateChain`. Sostituire i caratteri \n con nuove righe. Di seguito è riportata la struttura del file dopo aver incollato al suo interno il certificato e la catena di certificati.

```
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
```

11. Eseguire il comando seguente sul computer client per aggiungere questo certificato al keystore in modo da poterlo presentare quando si parla con i broker MSK.

```
keytool -keystore kafka.client.keystore.jks -import -file signed-certificate-from-
acm -alias Example-Alias -storepass Your-Store-Pass -keypass Your-Key-Pass
```

12. Crea un file denominato `client.properties` con i seguenti contenuti. Regolare le posizioni del truststore e del keystore sui percorsi in cui è stato salvato `kafka.client.truststore.jks`. Sostituisci i segnaposto con la versione del tuo client Kafka. *{YOUR KAFKA VERSION}*

```
security.protocol=SSL
ssl.truststore.location=/tmp/kafka_2.12-{YOUR KAFKA VERSION}/
kafka.client.truststore.jks
ssl.keystore.location=/tmp/kafka_2.12-{YOUR KAFKA VERSION}/
kafka.client.keystore.jks
ssl.keystore.password=Your-Store-Pass
ssl.key.password=Your-Key-Pass
```

Produci e consuma messaggi utilizzando l'autenticazione

Questo processo descrive come produrre e utilizzare messaggi utilizzando l'autenticazione.

1. Eseguire il comando seguente per creare un argomento. Il file denominato `client.properties` è quello creato nella procedura precedente.

```
<path-to-your-kafka-installation>/bin/kafka-topics.sh --create --bootstrap-
server BootstrapBroker-String --replication-factor 3 --partitions 1 --topic
 ExampleTopic --command-config client.properties
```

2. Eseguire il comando seguente per avviare un produttore della console. Il file denominato `client.properties` è quello creato nella procedura precedente.

```
<path-to-your-kafka-installation>/bin/kafka-console-producer.sh --bootstrap-
server BootstrapBroker-String --topic ExampleTopic --producer.config
 client.properties
```

3. In una nuova finestra di comando sul computer client, eseguire il comando seguente per avviare un consumatore della console.

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBroker-String --topic ExampleTopic --consumer.config
 client.properties
```

4. Digitare i messaggi nella finestra del produttore e guardarli apparire nella finestra del consumatore.

Autenticazione delle credenziali di accesso con Secrets Manager AWS

Puoi controllare l'accesso ai tuoi cluster Amazon MSK utilizzando credenziali di accesso archiviate e protette tramite Secrets Manager. AWS L'archiviazione delle credenziali utente in Secrets Manager riduce il sovraccarico dell'autenticazione del cluster, ad esempio il controllo, l'aggiornamento e la rotazione delle credenziali. Secrets Manager consente inoltre di condividere le credenziali utente tra i cluster.

Dopo aver associato un segreto a un cluster MSK, MSK sincronizza periodicamente i dati delle credenziali.

Questo argomento contiene le sezioni seguenti:

- Come funziona l'autenticazione delle credenziali di accesso
- Configurare SASL/SCRAM l'autenticazione per un cluster Amazon MSK
- Operazioni con gli utenti
- Limitazioni nell'uso dei segreti SCRAM

Come funziona l'autenticazione delle credenziali di accesso

L'autenticazione delle credenziali di accesso per Amazon MSK utilizza l'autenticazione SASL/SCRAM (Simple Authentication and Security Layer/ Salted Challenge Response Mechanism). Per configurare

l'autenticazione delle credenziali di accesso per un cluster, crea una risorsa segreta in AWS Secrets Manager e associa le credenziali di accesso a quel segreto.

SASL/SCRAM è definito in RFC 5802. SCRAM utilizza algoritmi di hashing protetti e non trasmette credenziali di accesso non crittografate tra client e server.

> **Note**
>
> Quando configuri SASL/SCRAM l'autenticazione per il tuo cluster, Amazon MSK attiva la crittografia TLS per tutto il traffico tra clienti e broker.

Configurare SASL/SCRAM l'autenticazione per un cluster Amazon MSK

Per impostare un segreto in AWS Secrets Manager, segui il tutorial Creazione e recupero di un segreto nella Guida per l'utente di AWS Secrets Manager.

Tieni presente i seguenti requisiti quando crei un segreto per un cluster Amazon MSK:

- Per il tipo di segreto, scegli Altro tipo di segreto (es. chiave API).
- Il nome del segreto deve iniziare con il prefisso AmazonMSK_.
- È necessario utilizzare una AWS KMS chiave personalizzata esistente o creare una nuova AWS KMS chiave personalizzata per il segreto. Secrets Manager utilizza la AWS KMS chiave predefinita per un segreto per impostazione predefinita.

  > **Important**
  >
  > Un segreto creato con la AWS KMS chiave predefinita non può essere utilizzato con un cluster Amazon MSK.

- I dati delle credenziali di accesso devono essere nel seguente formato per inserire coppie chiave-valore utilizzando l'opzione Non crittografato.

  ```
  {
    "username": "alice",
    "password": "alice-secret"
  }
  ```

- Prendi nota del valore del nome della risorsa Amazon (ARN) del segreto.

- > **Important**
  >
  > Non è possibile associare un segreto di Secrets Manager a un cluster che supera i limiti descritti in the section called “ Dimensionamento corretto del cluster: numero di partizioni per broker standard”.

- Se si utilizza il AWS CLI per creare il segreto, specificare un ID chiave o un ARN per il `kms-key-id` parametro. Non specificare un alias.
- Per associare il segreto al cluster, utilizza la console Amazon MSK o l' BatchAssociateScramSecretoperazione.

  > **Important**
  >
  > Quando associ un segreto a un cluster, Amazon MSK collega al segreto una policy delle risorse che consente al cluster di accedere e leggere i valori del segreto che hai definito. Questa policy delle risorse non dovrebbe essere modificata. In questo modo, è possibile impedire al cluster di accedere al segreto. Se apporti modifiche alla policy delle risorse Secrets e/o alla chiave KMS utilizzata per la crittografia segreta, assicurati di riassociare i segreti al tuo cluster MSK. In questo modo il cluster potrà continuare ad accedere al segreto.

  L'esempio seguente di input JSON per l'operazione `BatchAssociateScramSecret` associa un segreto a un cluster:

```
{
  "clusterArn" : "arn:aws:kafka:us-west-2:0123456789019:cluster/SalesCluster/
abcd1234-abcd-cafe-abab-9876543210ab-4",
  "secretArnList": [
    "arn:aws:secretsmanager:us-west-2:0123456789019:secret:AmazonMSK_MyClusterSecret"
  ]
}
```

Connessione al cluster con credenziali di accesso

Dopo aver creato un segreto e averlo collegato al cluster, è possibile collegare il client al cluster. La procedura seguente mostra come connettere un client a un cluster che utilizza SASL/SCRAM

l'autenticazione. Viene inoltre illustrato come produrre e consumare partendo da un argomento di esempio.

Argomenti

Connessione di un client al cluster tramite SASL/SCRAM l'autenticazione

1. Esegui il comando seguente su un computer che è stato AWS CLI installato. Sostituisci *clusterARN* con l'ARN del tuo cluster.

   ```
   aws kafka get-bootstrap-brokers --cluster-arn clusterARN
   ```

   Dal risultato JSON di questo comando, salva il valore associato alla stringa denominata. `BootstrapBrokerStringSaslScram` Utilizzerai questo valore nei passaggi successivi.
2. Sul tuo computer client, crea un file di configurazione JAAS che contenga le credenziali utente archiviate nel tuo segreto. Ad esempio, per l'utente alice, crea un file chiamato `users_jaas.conf` con il seguente contenuto.

   ```
   KafkaClient {
      org.apache.kafka.common.security.scram.ScramLoginModule required
      username="alice"
      password="alice-secret";
   };
   ```
3. Utilizza il seguente comando per esportare il file di configurazione JAAS come parametro di ambiente `KAFKA_OPTS`.

   ```
   export KAFKA_OPTS=-Djava.security.auth.login.config=<path-to-jaas-file>/
   users_jaas.conf
   ```
4. Nella directory `/tmp`, crea un file denominato `kafka.client.truststore.jks`.
5. (Facoltativo) Utilizzate il seguente comando per copiare il file dell'archivio chiavi JDK dalla `cacerts` cartella JVM nel `kafka.client.truststore.jks` file creato nel passaggio precedente. *JDKFolder*Sostituiscilo con il nome della cartella JDK sull'istanza. Ad esempio, la tua cartella JDK potrebbe avere il nome `java-1.8.0-openjdk-1.8.0.201.b09-0.amzn2.x86_64`.

```
cp /usr/lib/jvm/JDKFolder/lib/security/cacerts /tmp/kafka.client.truststore.jks
```

6. Nella directory `bin` di installazione di Apache Kafka, crea un file delle proprietà del client chiamato `client_sasl.properties` con il seguente contenuto. Questo file definisce il meccanismo e il protocollo SASL.

```
security.protocol=SASL_SSL
sasl.mechanism=SCRAM-SHA-512
```

7. Per creare un argomento di esempio, esegui il comando seguente. Sostituisci *BootstrapBrokerStringSaslScram* con la stringa del broker bootstrap ottenuta nel passaggio 1 di questo argomento.

```
<path-to-your-kafka-installation>/bin/kafka-topics.sh --create --bootstrap-
server BootstrapBrokerStringSaslScram --command-config <path-to-client-
properties>/client_sasl.properties --replication-factor 3 --partitions 1 --topic
 ExampleTopicName
```

8. Per produrre l'argomento di esempio che hai creato, esegui il comando seguente sul computer client. Sostituiscila *BootstrapBrokerStringSaslScram* con la stringa del broker bootstrap recuperata nel passaggio 1 di questo argomento.

```
<path-to-your-kafka-installation>/bin/kafka-console-producer.sh --broker-
list BootstrapBrokerStringSaslScram --topic ExampleTopicName --producer.config
 client_sasl.properties
```

9. Per utilizzare l'argomento che hai creato, esegui il comando seguente sul tuo computer client. Sostituiscila *BootstrapBrokerStringSaslScram* con la stringa del broker bootstrap ottenuta nel passaggio 1 di questo argomento.

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBrokerStringSaslScram --topic ExampleTopicName --from-beginning --
consumer.config client_sasl.properties
```

Risoluzione dei problemi di connessione

Quando si eseguono i comandi del client Kafka, è possibile riscontrare errori nella memoria heap di Java, specialmente quando si lavora con argomenti o set di dati di grandi dimensioni. Questi errori

si verificano perché gli strumenti Kafka vengono eseguiti come applicazioni Java con impostazioni di memoria predefinite che potrebbero essere insufficienti per il carico di lavoro.

Per risolvere `Out of Memory Java Heap` gli errori, è possibile aumentare la dimensione dell'heap Java modificando la variabile di `KAFKA_OPTS` ambiente per includere le impostazioni di memoria.

L'esempio seguente imposta la dimensione massima dell'heap su 1 GB (). `-Xmx1G` È possibile regolare questo valore in base alla memoria di sistema disponibile e ai requisiti.

```
export KAFKA_OPTS="-Djava.security.auth.login.config=<path-to-jaas-file>/
users_jaas.conf -Xmx1G"
```

Per approfondire argomenti di grandi dimensioni, prendi in considerazione l'utilizzo di parametri basati sul tempo o sull'offset anziché `--from-beginning` limitare l'utilizzo della memoria:

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBrokerStringSaslScram --topic ExampleTopicName --max-messages 1000 --
consumer.config client_sasl.properties
```

Operazioni con gli utenti

Creazione di utenti: crea utenti nel tuo segreto come coppie chiave-valore. Quando si utilizza l'opzione Non crittografato nella console Secrets Manager, è necessario specificare i dati delle credenziali di accesso nel formato seguente.

```
{
  "username": "alice",
  "password": "alice-secret"
}
```

Revoca dell'accesso utente: per revocare le credenziali di accesso a un cluster di un utente, si consiglia di rimuovere o applicare un'ACL al cluster e successivamente annullare l'associazione del segreto. Ciò può essere dovuto ai motivi seguenti:

- La rimozione di un utente non chiude le connessioni esistenti.
- La propagazione delle modifiche al segreto richiede fino a 10 minuti.

Per ulteriori informazioni sull'utilizzo delle ACL con Amazon MSK, consulta la pagina Apache Kafka ACLs.

Per i cluster che utilizzano ZooKeeper la modalità, si consiglia di limitare l'accesso ai ZooKeeper nodi per impedire agli utenti di apportare modifiche. ACLs Per ulteriori informazioni, consulta Controlla l'accesso ai ZooKeeper nodi Apache nel tuo cluster Amazon MSK.

Limitazioni nell'uso dei segreti SCRAM

Quando utilizzi i segreti SCRAM, tieni presente le limitazioni seguenti:

- Amazon MSK supporta solo l'autenticazione SCRAM-SHA-512.
- Un cluster Amazon MSK può avere fino a 1.000 utenti.
- Devi usare un AWS KMS key con il tuo segreto. Non è possibile utilizzare un segreto che utilizza la chiave di crittografia Secrets Manager predefinita con Amazon MSK. Per ulteriori informazioni sulla creazione di una chiave KMS, consulta la pagina Creating symmetric encryption KMS keys.
- Non è possibile utilizzare una chiave KMS asimmetrica con Secrets Manager.
- È possibile associare fino a 10 segreti a un cluster alla volta utilizzando l' BatchAssociateScramSecretoperazione.
- Il nome dei segreti associati a un cluster Amazon MSK deve avere il prefisso AmazonMSK_.
- I segreti associati a un cluster Amazon MSK devono trovarsi nello stesso account e nella stessa AWS regione Amazon Web Services del cluster.

Apache Kafka ACLs

Apache Kafka dispone di un autorizzatore collegabile e viene fornito con un'implementazione di autorizzazione. out-of-box Amazon MSK abilita questo provider di autorizzazioni nel file `server.properties` sui broker.

Apache Kafka ACLs ha il formato «Principal P è [Consentita/Negata] Operazione O dall'host H su qualsiasi risorsa R corrispondente a RP». ResourcePattern Se RP non corrisponde a una risorsa specifica R, allora R non ha alcun associato ACLs e quindi nessun altro oltre ai super utenti può accedere a R. Per modificare questo comportamento di Apache Kafka, impostate la proprietà su true. `allow.everyone.if.no.acl.found` In Amazon MSK è impostata su true per impostazione predefinita. Ciò significa che con i cluster Amazon MSK, se non ACLs imposti esplicitamente una risorsa, tutti i principali possono accedere a tale risorsa. Se abiliti una ACLs risorsa, solo i responsabili autorizzati possono accedervi. Se desideri limitare l'accesso a un argomento e autorizzare un client utilizzando l'autenticazione reciproca TLS, aggiungi ACLs utilizzando la CLI di autorizzazione Apache Kafka. Per ulteriori informazioni sull'aggiunta, la rimozione e l'elenco, consulta Kafka Authorization Command Line ACLs Interface.

Poiché Amazon MSK configura i broker come utenti privilegiati, possono accedere a tutti gli argomenti. Questo aiuta i broker a replicare i messaggi dalla partizione primaria indipendentemente dal fatto che la `allow.everyone.if.no.acl.found` proprietà sia definita o meno per la configurazione del cluster.

Per aggiungere o rimuovere l'accesso in lettura e scrittura a un argomento

1. Aggiungi i tuoi broker alla tabella ACL per consentire loro di leggere tutti gli argomenti esistenti. ACLs Per concedere ai broker l'accesso in lettura a un argomento, esegui il comando seguente su un computer client in grado di comunicare con il cluster MSK.

   Sostituiscila *`Distinguished-Name`* con il DNS di uno qualsiasi dei broker bootstrap del tuo cluster, quindi sostituisci la stringa prima del primo punto di questo nome distinto con un asterisco (). * Ad esempio, se uno dei broker di bootstrap del tuo cluster dispone del DNS`b-6.mytestcluster.67281x.c4.kafka.us-east-1.amazonaws.com`, sostituiscilo nel comando seguente con. *`Distinguished-Name`* `*.mytestcluster.67281x.c4.kafka.us-east-1.amazonaws.com` Per informazioni su come ottenere i broker bootstrap, consulta the section called “Ottieni i broker bootstrap”.

   ```
   <path-to-your-kafka-installation>/bin/kafka-acls.sh --bootstrap-server
    BootstrapServerString --add --allow-principal "User:CN=Distinguished-Name" --
   operation Read --group=* --topic Topic-Name
   ```

2. Per concedere a un'applicazione client l'accesso in lettura a un argomento, esegui il comando seguente sul computer client. Se utilizzi l'autenticazione TLS reciproca, utilizza la stessa *`Distinguished-Name`* che hai usato quando hai creato la chiave privata.

   ```
   <path-to-your-kafka-installation>/bin/kafka-acls.sh --bootstrap-server
    BootstrapServerString --add --allow-principal "User:CN=Distinguished-Name" --
   operation Read --group=* --topic Topic-Name
   ```

   Per rimuovere l'accesso in lettura, è possibile eseguire lo stesso comando, sostituendo `--add` con `--remove`.

3. Per concedere l'accesso in scrittura a un argomento, eseguire il comando seguente sul computer client. Se utilizzi l'autenticazione TLS reciproca, usa la stessa *`Distinguished-Name`* che hai usato quando hai creato la chiave privata.

```
<path-to-your-kafka-installation>/bin/kafka-acls.sh --bootstrap-server
 BootstrapServerString --add --allow-principal "User:CN=Distinguished-Name" --
operation Write --topic Topic-Name
```

Per rimuovere l'accesso in scrittura, è possibile eseguire lo stesso comando, sostituendo `--add` con `--remove`.

Modifica del gruppo di sicurezza di un cluster Amazon MSK

Questa pagina spiega come modificare il gruppo di sicurezza di un cluster MSK esistente. Potrebbe essere necessario modificare il gruppo di sicurezza di un cluster per fornire l'accesso a un determinato gruppo di utenti o per limitare l'accesso al cluster. Per ulteriori informazioni sui gruppi di sicurezza, consulta la pagina Security groups for your VPC nella Guida per l'utente di Amazon VPC.

1. Usa l'ListNodesAPI o il comando list-nodes in AWS CLI per ottenere un elenco dei broker del tuo cluster. I risultati di questa operazione includono le interfacce IDs di rete elastiche (ENIs) associate ai broker.
2. Accedi a AWS Management Console e apri la EC2 console Amazon all'indirizzo https://console.aws.amazon.com/ec2/.
3. Utilizzando l'elenco a discesa nell'angolo in alto a destra della schermata, seleziona la regione in cui è implementato il cluster.
4. Nel riquadro a sinistra, in Rete e sicurezza, scegli Interfacce di rete.
5. Seleziona il primo ENI che hai ottenuto nel primo passaggio. Scegli il menu Operazioni nella parte superiore dello schermo, quindi scegli Modifica gruppi di sicurezza. Assegna il nuovo gruppo di sicurezza a questo ENI. Ripeti questo passaggio per ognuno di ENIs quelli che hai ottenuto nel primo passaggio.

   > **Note**
   >
   > Le modifiche apportate al gruppo di sicurezza di un cluster utilizzando la EC2 console Amazon non si riflettono nella console MSK in Impostazioni di rete.

6. Configura le regole del nuovo gruppo di sicurezza per garantire che i tuoi client abbiano accesso ai broker. Per informazioni sull'impostazione delle regole del gruppi di sicurezza, consulta la pagina Adding, Removing, and Updating Rules nella guida per l'utente di Amazon VPC.

> **Important**
>
> Se modifichi il gruppo di sicurezza associato ai broker di un cluster e poi aggiungi nuovi broker a tale cluster, Amazon MSK associa i nuovi broker al gruppo di sicurezza originale associato al cluster al momento della creazione dello stesso. Tuttavia, affinché un cluster funzioni correttamente, tutti i relativi broker devono essere associati allo stesso gruppo di sicurezza. Pertanto, se aggiungi nuovi broker dopo aver modificato il gruppo di sicurezza, devi seguire nuovamente la procedura precedente e aggiornare i ENIs nuovi broker.

Controlla l'accesso ai ZooKeeper nodi Apache nel tuo cluster Amazon MSK

Per motivi di sicurezza, puoi limitare l'accesso ai ZooKeeper nodi Apache che fanno parte del tuo cluster Amazon MSK. Per limitare l'accesso ai nodi, puoi assegnare loro un gruppo di sicurezza separato. Puoi quindi stabilire chi ottiene l'accesso a tale gruppo di sicurezza.

> **Important**
>
> Questa sezione non si applica ai cluster in esecuzione in modalità. KRaft Consultare the section called “KRaft modalità ”.

Questo argomento contiene le sezioni seguenti:

- Per collocare i ZooKeeper nodi Apache in un gruppo di sicurezza separato
- Utilizzo della sicurezza TLS con Apache ZooKeeper

Per collocare i ZooKeeper nodi Apache in un gruppo di sicurezza separato

Per limitare l'accesso ai ZooKeeper nodi Apache, puoi assegnare loro un gruppo di sicurezza separato. Puoi scegliere chi ha accesso a questo nuovo gruppo di sicurezza impostando le regole del gruppo di sicurezza.

1. Ottieni la stringa di ZooKeeper connessione Apache per il tuo cluster. Per scoprire come, consulta the section called “ZooKeeper modalità”. La stringa di connessione contiene i nomi DNS dei tuoi nodi ZooKeeper Apache.

2. Utilizzare uno strumento simile a `host` o `ping` per convertire i nomi DNS ottenuti nel passaggio precedente in indirizzi IP. Salvare questi indirizzi IP perché saranno necessari più avanti in questa procedura.
3. Accedi a AWS Management Console e apri la EC2 console Amazon all'indirizzo https://console.aws.amazon.com/ec2/.
4. Nel riquadro di navigazione, in NETWORK & SECURITY (Rete e sicurezza), scegliere Network Interfaces (Interfacce di rete).
5. Nel campo di ricerca sopra la tabella delle interfacce di rete, digitare il nome del cluster, quindi premere Invio. Questo limita il numero di interfacce di rete visualizzate nella tabella a quelle associate al cluster.
6. Selezionare la casella di controllo all'inizio della riga corrispondente alla prima interfaccia di rete nell'elenco.
7. Nel riquadro dei dettagli in fondo alla pagina, cerca l' IPv4 IP privato primario. Se questo indirizzo IP corrisponde a uno degli indirizzi IP ottenuti nel primo passaggio di questa procedura, significa che questa interfaccia di rete è assegnata a un ZooKeeper nodo Apache che fa parte del cluster. In caso contrario, deselezionare la casella di controllo accanto a questa interfaccia di rete e selezionare l'interfaccia di rete successiva nell'elenco. L'ordine di selezione delle interfacce di rete non ha importanza. Nei passaggi successivi, eseguirai le stesse operazioni su tutte le interfacce di rete assegnate ai ZooKeeper nodi Apache, una per una.
8. Quando selezioni un'interfaccia di rete che corrisponde a un ZooKeeper nodo Apache, scegli il menu Azioni nella parte superiore della pagina, quindi scegli Cambia gruppi di sicurezza. Assegnare un nuovo gruppo di sicurezza a questa interfaccia di rete. Per ulteriori informazioni sulla creazione dei gruppi di sicurezza, consulta la pagina Creating a Security Group nella documentazione di Amazon VPC.
9. Ripeti il passaggio precedente per assegnare lo stesso nuovo gruppo di sicurezza a tutte le interfacce di rete associate ai ZooKeeper nodi Apache del cluster.
10. È ora possibile scegliere chi dispone dell'accesso a questo nuovo gruppo di sicurezza. Per informazioni sull'impostazione delle regole dei gruppi di sicurezza, consulta la pagina Adding, Removing, and Updating Rules nella documentazione di Amazon VPC.

Utilizzo della sicurezza TLS con Apache ZooKeeper

Puoi utilizzare la sicurezza TLS per la crittografia in transito tra i tuoi client e i tuoi nodi Apache. ZooKeeper Per implementare la sicurezza TLS con i ZooKeeper nodi Apache, procedi come segue:

- I cluster devono utilizzare Apache Kafka versione 2.5.1 o successiva per utilizzare la sicurezza TLS con Apache. ZooKeeper
- Abilita la sicurezza TLS quando crei o configuri il cluster. I cluster creati con Apache Kafka versione 2.5.1 o successiva con TLS abilitato utilizzano automaticamente la sicurezza TLS con gli endpoint Apache. ZooKeeper Per informazioni sulla configurazione della sicurezza TLS, consulta la pagina Inizia a usare la crittografia Amazon MSK.
- Recupera gli endpoint TLS Apache utilizzando l'operazione. ZooKeeper DescribeCluster
- Crea un file di ZooKeeper configurazione Apache da utilizzare con `kafka-acls.sh`gli strumenti `kafka-configs.sh` and o con la shell. ZooKeeper Con ogni strumento, si utilizza il `--zk-tls-config-file` parametro per specificare la configurazione di Apache ZooKeeper .

  L'esempio seguente mostra un tipico file di configurazione di Apache ZooKeeper :

  ```
  zookeeper.ssl.client.enable=true
  zookeeper.clientCnxnSocket=org.apache.zookeeper.ClientCnxnSocketNetty
  zookeeper.ssl.keystore.location=kafka.jks
  zookeeper.ssl.keystore.password=test1234
  zookeeper.ssl.truststore.location=truststore.jks
  zookeeper.ssl.truststore.password=test1234
  ```

- Per altri comandi (come`kafka-topics`), è necessario utilizzare la variabile di `KAFKA_OPTS` ambiente per configurare i parametri di Apache ZooKeeper. L'esempio seguente mostra come configurare la variabile di `KAFKA_OPTS` ambiente per passare i ZooKeeper parametri Apache ad altri comandi:

  ```
  export KAFKA_OPTS="
  -Dzookeeper.clientCnxnSocket=org.apache.zookeeper.ClientCnxnSocketNetty
  -Dzookeeper.client.secure=true
  -Dzookeeper.ssl.trustStore.location=/home/ec2-user/kafka.client.truststore.jks
  -Dzookeeper.ssl.trustStore.password=changeit"
  ```

  Dopo aver configurato la variabile di ambiente `KAFKA_OPTS`, è possibile utilizzare normalmente i comandi della CLI. L'esempio seguente crea un argomento di Apache Kafka utilizzando la ZooKeeper configurazione di Apache dalla variabile di ambiente: `KAFKA_OPTS`

  ```
  <path-to-your-kafka-installation>/bin/kafka-topics.sh --create --
  zookeeper ZooKeeperTLSConnectString --replication-factor 3 --partitions 1 --topic
   AWSKafkaTutorialTopic
  ```

> **Note**
>
> I nomi dei parametri utilizzati nel file di ZooKeeper configurazione di Apache e quelli utilizzati nella variabile di `KAFKA_OPTS` ambiente non sono coerenti. Presta attenzione ai nomi che usi con ciascun parametri nel file di configurazione e nella variabile di ambiente `KAFKA_OPTS`.

Per ulteriori informazioni sull'accesso ai ZooKeeper nodi Apache con TLS, vedi KIP-515: Abilita il client ZK a usare la nuova autenticazione supportata da TLS.

Convalida della conformità per Streaming gestito da Amazon per Apache Kafka

Revisori di terze parti valutano la sicurezza e la conformità di Streaming gestito da Amazon per Apache Kafka nell'ambito dei programmi di conformità di AWS . Questi includono PCI e HIPAA BAA.

Per un elenco di AWS servizi nell'ambito di programmi di conformità specifici, consulta Amazon Services in Scope by Compliance Program . Per informazioni generali, consulta Programmi di AWS conformità Programmi di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La tua responsabilità di conformità quando usi Amazon MSK è determinata dalla sensibilità dei tuoi dati, dagli obiettivi di conformità della tua azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Security and Compliance Quick Start Guides (Guide Quick Start Sicurezza e compliance): queste guide alla distribuzione illustrano considerazioni relative all'architettura e forniscono procedure per la distribuzione di ambienti di base incentrati sulla sicurezza e sulla conformità su AWS.
- Whitepaper sull'architettura per la sicurezza e la conformità HIPAA: questo white paper descrive come le aziende possono utilizzare per creare applicazioni conformi allo standard HIPAA. AWS
- AWS Risorse per la conformità Risorse per la conformità: questa raccolta di potrebbe riguardare il settore e la località in cui operate.
- Valutazione delle risorse con le regole nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.

- AWS Security Hub— Questo AWS servizio offre una visione completa dello stato di sicurezza dell'utente e consente di verificare la conformità agli standard e alle best practice del settore della sicurezza. AWS

Resilienza in Streaming gestito da Amazon per Apache Kafka

L'infrastruttura AWS globale è costruita attorno a regioni e zone di disponibilità. AWS AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta Global Infrastructure.AWS

Sicurezza dell'infrastruttura in Streaming gestito da Amazon per Apache Kafka

In quanto servizio gestito, Amazon Managed Streaming for Apache Kafka è protetto AWS dalle procedure di sicurezza di rete globali descritte nel white paper di Amazon Web Services: Overview of Security Processes.

Utilizzi chiamate API AWS pubblicate per accedere ad Amazon MSK attraverso la rete. I client devono supportare Transport Layer Security (TLS) 1.0 o versioni successive. È consigliabile TLS 1.2 o versioni successive. I client devono, inoltre, supportare le suite di cifratura con PFS (Perfect Forward Secrecy), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Configurazione Amazon MSK Provisioned

Amazon MSK fornisce configurazioni predefinite per broker, argomenti e nodi di metadati. Puoi inoltre creare configurazioni personalizzate e utilizzarle per creare nuovi cluster MSK o per aggiornare cluster esistenti. Una configurazione MSK è costituita da un insieme di proprietà e dai relativi valori corrispondenti. A seconda del tipo di broker utilizzato nel cluster, esistono diversi set di impostazioni

predefinite di configurazione e un diverso set di configurazioni che è possibile modificare. Consulta le sezioni seguenti per maggiori dettagli su come configurare i broker Standard ed Express.

Argomenti

- Configurazioni standard del broker
- Configurazioni del broker Express
- Operazioni di configurazione del broker

Configurazioni standard del broker

Questa sezione descrive le proprietà di configurazione per i broker Standard.

Argomenti

- Configurazioni Amazon MSK personalizzate
- Configurazione Amazon MSK predefinita
- Linee guida per la configurazione a livello di argomento dello storage su più livelli di Amazon MSK

Configurazioni Amazon MSK personalizzate

Puoi utilizzare Amazon MSK per creare una configurazione MSK personalizzata in cui vengono impostate le proprietà seguenti. Le proprietà che non vengono impostate in modo esplicito ricevono i valori che hanno in the section called “Configurazione Amazon MSK predefinita”. Per ulteriori informazioni sulle proprietà di configurazione, consulta la pagina relativa alla configurazione di Apache Kafka.

Proprietà di configurazione di Apache Kafka

Nome	Descrizione
allow.everyone.if.no.acl.found	Se desideri impostare questa proprietà su`false`, assicurati innanzitutto di definire Apache Kafka ACLs per il tuo cluster. Se imposti questa proprietà su `false` e non definisci prima Apache Kafka ACLs, perdi l'accesso al cluster. In tal caso, puoi aggiornar e nuovamente la configurazione e impostare

Nome	Descrizione
	questa proprietà su `true` per ottenere nuovamente l'accesso al cluster.
auto.create.topics.enable	Abilita la creazione automatica di argomenti sul server.
compression.type	Il tipo di compressione finale per un determinato argomento. Puoi impostare questa proprietà sui codec di compressione standard (`gzip`, `snappy`, `lz4` e `zstd`). Inoltre, accetta `uncompressed` . Questo valore equivale a nessuna compressione. Se imposti il valore su `producer`, significa mantenere il codec di compressione originale impostato dal produttore.
connections.max.idle.ms	Timeout delle connessioni inattive in millisecondi. I thread del processore dei socket del server chiudono le connessioni inattive per un periodo superiore al valore impostato per questa proprietà.
default.replication.factor	Il fattore di replica predefinito per argomenti creati automaticamente.
delete.topic.enable	Abilita l'operazione di eliminazione argomento . Se questa configurazione è disattivata, non è possibile eliminare un argomento tramite lo strumento di amministrazione.

Nome	Descrizione
group.initial.rebalance.delay.ms	Il tempo durante il quale il coordinatore del gruppo attende che altri consumatori si uniscano a un nuovo gruppo prima di eseguire il primo ribilanciamento. Un ritardo più lungo significa potenzialmente meno ribilanciamenti, ma aumenta il tempo prima dell'inizio dell'elaborazione.
group.max.session.timeout.ms	Timeout sessione massimo per i consumatori registrati. Timeout più lunghi offrono ai consumatori più tempo per elaborare i messaggi tra heartbeat, ma implicano un aumento del tempo richiesto per rilevare gli errori.
group.min.session.timeout.ms	Timeout sessione minimo per consumatori registrati. Timeout più brevi comportano un rilevamento più rapido degli errori, ma implicano un heartbeat dei consumatori più frequente. Ciò può sovraccaricare le risorse del broker.
leader.imbalance.per.broker.percentage	Il rapporto di squilibrio leader consentito per broker. Il controller attiva un bilanciamento dei leader se supera questo valore per broker. Questo valore è specificato in percentuale.
log.cleaner.delete.retention.ms	Quantità di tempo per cui Apache Kafka deve conservare i record eliminati. Il valore minimo è 0.

Nome	Descrizione
log.cleaner.min.cleanable.ratio	Questa proprietà di configurazione può avere valori compresi tra 0 e 1. Questo valore determina la frequenza con cui il compattatore di log tenta di pulire il log (se la compattazione dei log è abilitata). Per impostazione predefini ta, Apache Kafka evita di pulire un log se più del 50% del log è stato compattato. Questo rapporto limita lo spazio massimo che il log spreca con i duplicati (al 50%, ciò significa che al massimo il 50% del log potrebbe essere duplicato). Un rapporto più elevato significa un numero inferiore, pulizie più efficienti, ma anche più spreco di spazio nel log.
log.cleanup.policy	La policy di pulizia predefinita per i segmenti oltre la finestra di conservazione. Un elenco separato da virgole di policy valide. Policy valide sono `delete` e `compact`. Per i cluster abilitati all'archiviazione a più livelli, è valida solo la policy `delete`.
log.flush.interval.messages	Numero di messaggi che si accumulano in una partizione di log prima che i messaggi vengano scaricati su disco.
log.flush.interval.ms	Tempo massimo, in millisecondi, di mantenime nto in memoria di un messaggio in un argomento prima che venga scaricato su disco. Se questo valore non viene impostato, viene utilizzato il valore in log.flush.scheduler.interva l.ms. Il valore minimo è 0.

Nome	Descrizione
log.message.timestamp.difference.max.ms	Questa configurazione è obsoleta in Kafka 3.6.0. Sono state aggiunte due configurazioni, e. `log.message.timestamp.before.max.ms` `log.message.timestamp.after.max.ms` La differenza massima consentita tra il timestamp del momento in cui un broker riceve un messaggio e il timestamp specificato nel messaggio. Se log.messa ge.timestamp.type=CreateTime, un messaggio viene rifiutato se la differenza di timestamp supera questa soglia. Questa LogAppendTime configurazione viene ignorata se log.messa ge.timestamp.type=.
log.message.timestamp.type	Specifica se il timestamp nel messaggio è l'ora di creazione del messaggio o l'ora di aggiunta del log. I valori consentiti sono `CreateTime` e `LogAppendTime` .
log.retention.bytes	Dimensione massima del log prima dell'elim inazione.
log.retention.hours	Numero di ore per cui mantenere un file di log prima di eliminarlo, terziario alla proprietà log.retention.ms.
log.retention.minutes	Numero di minuti per cui mantenere un file di log prima di eliminarlo, secondario alla proprietà log.retention.ms. Se questo valore non viene impostato, viene utilizzato il valore in log.retention.hours.
log.retention.ms	Numero di millisecondi per cui mantenere un file di log prima di eliminarlo. Se non è impostato, viene utilizzato il valore in log.reten tion.minutes.

Nome	Descrizione
log.roll.ms	Tempo massimo prima che un nuovo segmento di log venga distribuito (in millisecondi). Se questo valore non viene impostato, viene utilizzato il valore in log.roll.hours. Il valore minimo possibile per questa proprietà è 1.
log.segment.bytes	Dimensione massima di un singolo file di log.
max.incremental.fetch.session.cache.slots	Numero massimo di sessioni di recupero incrementali che vengono mantenute.
message.max.bytes	Dimensione massima del batch di record consentita da Kafka. Se aumenti questo valore e sono presenti consumatori più vecchi di 0.10.2, anche le dimensioni di recupero dei consumatori devono essere incrementate in modo che possano recuperare batch di record di queste dimensioni. Nella versione più recente del formato del messaggio, i messaggi vengono sempre raggruppati in batch per maggiore efficienz a. Nelle versioni precedenti del formato del messaggio, i record non compressi non sono raggruppati in batch; in tal caso, questo limite si applica solo a un singolo record. Questo valore può essere impostato a livello di argomento con la configurazione max.messa ge.bytes.

Nome	Descrizione
min.insync.replicas	Quando un produttore imposta le ACK su `"all"` (o `"-1"`), il valore in min.insync.replica s specifica il numero minimo di repliche che devono riconoscere una scrittura affinché questa sia considerata correttamente completat a. Se questo minimo non può essere raggiunto , il produttore solleva un'eccezione (o). NotEnoughReplicas NotEnoughReplicasA fterAppend È possibile utilizzare i valori in min.insyn c.replicas e ACK per applicare maggiori garanzie di durabilità. Ad esempio, è possibile creare un argomento con un fattore di replica di 3, impostare min.insync.replicas su 2 e produrre con ACK di `"all"`. Ciò garantisce che il produttore generi un'eccezione se la maggior parte delle repliche non riceve una scrittura.
num.io.thread	Il numero di thread utilizzati dal server per elaborare le richieste, che possono includere I/ O del disco.
num.network.threads	Il numero di thread utilizzati dal server per ricevere richieste dalla rete e inviarle le risposte.
num.partitions	Numero predefinito di partizioni di log per argomento.
num.recovery.threads.per.data.dir	Il numero di thread per directory di dati da utilizzare per il ripristino dei log all'avvio e per lo scaricamento all'arresto.

Nome	Descrizione
num.replica.fetchers	Il numero di thread fetcher utilizzati per rispondere ai messaggi da un broker di origine. Se aumenti questo valore, puoi aumentare il grado di I/O parallelismo nel broker di follower.
offsets.retention.minutes	Dopo che un gruppo di consumatori perde tutti i suoi consumatori (ovvero, diventa vuoto) i suoi offset vengono mantenuti per questo periodo di conservazione prima di essere scartati. Per i consumatori autonomi (ossia che utilizzano l'assegnazione manuale), gli offset scadono dopo l'ora dell'ultimo commit più questo periodo di conservazione.
offsets.topic.replication.factor	Il fattore di replica per l'argomento di offset. La selezione di un valore più alto garantisce la disponibilità. La creazione di argomenti interni non riesce fino a quando la dimensione del cluster non soddisfa questo requisito del fattore di replica.
replica.fetch.max.bytes	Numero di byte di messaggi da recuperare per ogni partizione. Questo valore non è un massimo assoluto. Se il primo batch di record nella prima partizione non vuota del recupero è più grande di questo valore, viene restituito il batch di record per garantire l'avanzamento. La proprietà message.max.bytes (configurazione broker) o max.message.bytes (configurazione argomento) specifica la dimensione massima del batch di record accettata dal broker.

Nome	Descrizione
replica.fetch.response.max.bytes	Il numero massimo di byte previsto per l'intera risposta di recupero. I record vengono recuperati in batch. Se il primo batch di record nella prima partizione non vuota del recupero è più grande di questo valore, il batch di record verrà comunque restituito per garantire l'avanzamento. Questo non è un massimo assoluto. Le proprietà message.max.bytes (configurazione broker) o max.message.bytes (configurazione argomento) specificano la dimensione massima del batch di record accettata dal broker.
replica.lag.time.max.ms	Se un follower non ha inviato richieste di fetch o non ha consumato fino all'offset di fine log del leader per almeno questo numero di milliseco ndi, il leader rimuove il follower dall'ISR. MinValue: 10000 MaxValue = 30000
replica.selector.class	Il nome completo della classe che implement a. ReplicaSelector Il broker utilizza questo valore per trovare la replica di lettura preferita . Se utilizzi Apache Kafka versione 2.4.1 o superiore e desideri consentire ai consumato ri di recuperare dati dalla replica più vicina, imposta questa proprietà su `org.apach e.kafka.common.replica.Rack AwareReplicaSelector` . Per ulteriori informazioni, consulta the section called “Apache Kafka versione 2.4.1 (usa invece 2.4.1.1)”.

Nome	Descrizione
replica.socket.receive.buffer.bytes	Il buffer di ricezione socket per le richieste di rete.
socket.receive.buffer.bytes	Buffer SO_RCVBUF dei socket del server dei socket. Il valore minimo che è possibile impostare per questa proprietà è -1. Se il valore è -1, Amazon MSK utilizza il sistema operativo predefinito.
socket.request.max.bytes	Il numero massimo di byte in una richiesta socket.
socket.send.buffer.bytes	Buffer SO_SNDBUF dei socket del server dei socket. Il valore minimo che è possibile impostare per questa proprietà è -1. Se il valore è -1, Amazon MSK utilizza il sistema operativo predefinito.
transaction.max.timeout.ms	Timeout massimo per transazioni. Se il tempo di transazione richiesto da un cliente supera questo valore, il broker restituisce un errore in. InitProducerIdRequest Ciò evita un timeout troppo elevato per un client, che può rallentar e i consumatori che leggono dagli argomenti inclusi nella transazione.
transaction.state.log.min.isr	Configurazione min.insync.replicas ignorata per l'argomento di transazione.
transaction.state.log.replication.factor	Il fattore di replica per l'argomento di transazio ne. La selezione di un valore più alto per questa proprietà aumenta la disponibilità. La creazione di argomenti interni non riesce fino a quando la dimensione del cluster non soddisfa questo requisito del fattore di replica.

Nome	Descrizione
transactional.id.expiration.ms	Il tempo in millisecondi durante il quale il coordinatore della transazione attende di ricevere eventuali aggiornamenti sullo stato delle transazioni per la transazione corrente prima che il coordinatore faccia scadere il proprio ID transazionale. Questa impostazione influenza anche la scadenza dell'ID del produttore perché fa IDs scadere il produttore quando questo tempo trascorre dopo l'ultima scrittura con l'ID produttore specificato. Producer IDs potrebbe scadere prima se l'ultima scrittura dall'ID produttore viene eliminata a causa delle impostazioni di conservazione dell'argomento. Il valore minimo per questa proprietà è 1 millisecondo.
unclean.leader.election.enable	Indica se le repliche non incluse nel set ISR devono fungere da leader come ultima risorsa, anche se ciò potrebbe comportare la perdita di dati.
zookeeper.connection.timeout.ms	ZooKeeper cluster di modalità. Tempo massimo di attesa del client per stabilire una connessione. ZooKeeper Se questo valore non viene impostato, viene utilizzato il valore fornito in zookeeper.session.timeout.ms. MinValue = 6000 MaxValue (incluso) = 18000 Ti consigliamo di impostare questo valore su 10.000 su T3.small per evitare tempi di inattività del cluster.

Nome	Descrizione
zookeeper.session.timeout.ms	ZooKeeper cluster di modalità. Il timeout della ZooKeeper sessione Apache in millisecondi. MinValue = 6000 MaxValue (incluso) = 18000

Per informazioni su come creare una configurazione MSK personalizzata, elencare tutte le configurazioni o descriverle, consulta the section called “Operazioni di configurazione del broker”. Per creare un cluster MSK utilizzando una configurazione MSK personalizzata o per aggiornare un cluster con una nuova configurazione personalizzata, consulta la pagina the section called “Caratteristiche e concetti chiave”.

Quando si aggiorna il cluster MSK esistente con una configurazione MSK personalizzata, Amazon MSK esegue riavvii in sequenza quando necessario e utilizza le best practice per ridurre al minimo i tempi di inattività del cliente. Ad esempio, dopo aver riavviato ogni broker, Amazon MSK prova a lasciare che il broker recuperi i dati che potrebbe aver perso durante l'aggiornamento della configurazione prima di passare al broker successivo.

Configurazione dinamica di Amazon MSK

Oltre alle proprietà di configurazione fornite da Amazon MSK, puoi impostare dinamicamente le proprietà di configurazione a livello di cluster e broker che non richiedono un riavvio del broker. È possibile impostare dinamicamente alcune proprietà di configurazione. Si tratta delle proprietà che non sono contrassegnate come di sola lettura nella tabella in Broker Configs nella documentazione di Apache Kafka. Per informazioni sulla configurazione dinamica e sui comandi di esempio, consulta la pagina Updating Broker Configs nella documentazione di Apache Kafka.

> **Note**
>
> Puoi impostare la proprietà `advertised.listeners`, ma non la proprietà `listeners`.

Configurazione Amazon MSK a livello di argomento

Puoi utilizzare i comandi Apache Kafka per impostare o modificare le proprietà di configurazione a livello dell'argomento per argomenti nuovi ed esistenti. Per ulteriori informazioni sulle proprietà di

configurazione a livello di argomento ed esempi di come impostarle, consulta la pagina Topic-Level Configs nella documentazione ufficiale di Apache Kafka.

Configurazione Amazon MSK predefinita

Quando crei un cluster MSK senza specificare una configurazione MSK personalizzata, Amazon MSK crea e utilizza una configurazione predefinita con i valori visualizzati nella tabella seguente. Per le proprietà non presenti in questa tabella, Amazon MSK utilizza i valori predefiniti associati alla versione di Apache Kafka. Per un elenco di questi valori predefiniti, consulta la pagina relativa alla configurazione di Apache Kafka.

Valori di configurazione predefiniti

Nome	Descrizione	Valore predefinito per il cluster con l'archiviazione non a più livelli	Valore predefinito per il cluster abilitato all'archiviazione a più livelli
allow.everyone.if.no.acl.found	Se nessun modello di risorsa corrisponde a una risorsa specifica, la risorsa non è associata ACLs. In questo caso, se questa proprietà è impostata su `true`, tutti possono accedere alla risorsa, non solo i superutenti.	`true`	`true`
auto.create.topics.enable	Abilita la creazione automatica di un argomento sul server.	`false`	`false`
auto.leader.rebalance.enable	Consente il bilanciamento leader automatico. Un thread in background controlla e attiva il bilanciam	`true`	`true`

Nome	Descrizione	Valore predefinito per il cluster con l'archiviazione non a più livelli	Valore predefinito per il cluster abilitato all'archiviazione a più livelli
	ento del leader a intervalli regolari, se necessario.		
default.replicatio n.factor	Fattori di replica predefiniti per argomenti creati automaticamente.	3 per i cluster in 3 zone di disponibilità e 2 per i cluster in 2 zone di disponibilità.	3 per i cluster in 3 zone di disponibilità e 2 per i cluster in 2 zone di disponibilità.

Nome	Descrizione	Valore predefinito per il cluster con l'archiviazione non a più livelli	Valore predefinito per il cluster abilitato all'archiviazione a più livelli
local.retention.bytes	La dimensione massima dei segmenti di log locali per una partizione prima dell'eliminazione dei vecchi segmenti. Se questo valore non viene impostato, viene utilizzato il valore in log.retention.bytes. Il valore effettivo deve essere sempre minore o uguale al valore di log.retention.bytes. Il valore predefinito -2 indica che non è previsto un limite alla conservazione locale. Ciò corrisponde all'impostazione retention.ms/bytes di -1. Le proprietà local.retention.ms e local.retention.bytes sono simili a log.retention, in quanto vengono utilizzate per determinare per quanto tempo i segmenti di log devono rimanere	-2 per illimitato	-2 per illimitato

Nome	Descrizione	Valore predefinito per il cluster con l'archivi azione non a più livelli	Valore predefinito per il cluster abilitato all'archiviazione a più livelli
	nell'archiviazione locale. Le configura zioni log.reten tion.* esistenti sono configurazioni di conservazione per la partizione degli argomenti. Ciò include l'archiviazione locale e remota. Valori validi: numeri interi in [-2; +Inf]		

Nome	Descrizione	Valore predefinito per il cluster con l'archiviazione non a più livelli	Valore predefinito per il cluster abilitato all'archiviazione a più livelli
local.retention.ms	Numero di millisecondi di conservazione del segmento di log locale prima dell'eliminazione. Se questo valore non viene impostato, Amazon MSK utilizza il valore in log.retention.ms. Il valore effettivo deve essere sempre minore o uguale al valore di log.retention.bytes. Il valore predefinito -2 indica che non è previsto un limite alla conservazione locale. Ciò corrisponde all'impostazione retention.ms/bytes di -1. I valori local.retention.ms e local.retention.bytes sono simili a log.retention. MSK utilizza questa configurazione per determinare per quanto tempo i segmenti di log devono rimanere	-2 per illimitato	-2 per illimitato

Nome	Descrizione	Valore predefinito per il cluster con l'archivi azione non a più livelli	Valore predefinito per il cluster abilitato all'archiviazione a più livelli
	nell'archiviazione locale. Le configura zioni log.reten tion.* esistenti sono configurazioni di conservazione per la partizione degli argomenti. Ciò include l'archiviazione locale e remota. I valori validi sono numeri interi maggiori di 0.		

Nome	Descrizione	Valore predefinito per il cluster con l'archiviazione non a più livelli	Valore predefinito per il cluster abilitato all'archiviazione a più livelli
log.message.timestamp.difference.max.ms	Questa configurazione è obsoleta in Kafka 3.6.0. Sono state aggiunte due configurazioni, e. `log.message.timestamp.before.max.ms` `log.message.timestamp.after.max.ms` La differenza massima consentita tra il timestamp quando un broker riceve un messaggio e il timestamp specificato nel messaggio. Se log.message.timestamp.type=CreateTime, un messaggio verrà rifiutato se la differenza di timestamp supera questa soglia. Questa configurazione viene ignorata se log.message.timestamp.type=. LogAppendTime La differenza di	9223372036854775807	86400000 per Kafka 2.8.2.tiered e Kafka 3.7.x a più livelli.

Nome	Descrizione	Valore predefinito per il cluster con l'archiviazione non a più livelli	Valore predefinito per il cluster abilitato all'archiviazione a più livelli
	timestamp massima consentita non deve essere maggiore di log.retention.ms per evitare una distribuzione dei log inutilmente frequente.		
log.segment.bytes	Dimensione massima di un singolo file di log.	1073741824	134217728

Nome	Descrizione	Valore predefinito per il cluster con l'archiviazione non a più livelli	Valore predefinito per il cluster abilitato all'archiviazione a più livelli
min.insync.replicas	Quando un produttore imposta il valore delle ACK (il riconoscimento che il produttore riceve dal broker Kafka) su `"all"` (o `"-1"`), il valore in min.insync.replicas specifica il numero minimo di repliche che devono riconoscere una scrittura affinché questa sia considerata correttamente completata. Se questo valore non soddisfa questo minimo, il produttore solleva un'eccezione (o). NotEnoughReplicas NotEnoughReplicasAfterAppend Se usati insieme, i valori min.insync.replicas e ACK consentono di imporre maggiori garanzie di durata. Ad esempio, è possibile creare un argomento con un fattore di replica di 3,	2 per i cluster in 3 zone di disponibilità e 1 per i cluster in 2 zone di disponibilità.	2 per i cluster in 3 zone di disponibilità e 1 per i cluster in 2 zone di disponibilità.

Nome	Descrizione	Valore predefinito per il cluster con l'archiviazione non a più livelli	Valore predefinito per il cluster abilitato all'archiviazione a più livelli
	impostare min.insync.replicas su 2 e produrre con ACK di `"all"`. Ciò garantisce che il produttore generi un'eccezione se la maggior parte delle repliche non riceve una scrittura.		
num.io.thread	Numero di thread utilizzati dal server per produrre le richieste, che possono includere l'I/O del disco.	8	max (8, vCPUs) dove v CPUs dipende dalla dimensione dell'istanza del broker
num.network.threads	Numero di thread utilizzati dal server per ricevere richieste dalla rete e inviare risposte alla rete.	5	max (5, vCPUs /2) dove v CPUs dipende dalla dimensione dell'istanza del broker
num.partitions	Numero predefinito di partizioni di log per argomento.	1	1

Nome	Descrizione	Valore predefinito per il cluster con l'archivi azione non a più livelli	Valore predefinito per il cluster abilitato all'archiviazione a più livelli
num.replica.fetchers	Numero di thread fetcher utilizzati per replicare i messaggi da un broker di origine. Se si aumenta questo valore, è possibile aumentare il grado di paralleli smo nel broker di I/O follower.	2	max (2, vCPUs /4) dove v dipende dalla dimensione dell'ista nza del broker CPUs
remote.log.msk.dis able.policy	Utilizzato con remote.storage.ena ble per disabilitare l'archiviazione a più livelli. Imposta questa policy su Elimina per indicare che i dati nell'archiviazione a più livelli vengono eliminati quando si imposta remote.st orage.enable su false.	N/D	Nessuno
remote.log.reader. threads	La dimensione del pool di thread del lettore di log remoto, utilizzato nella pianificazione delle attività per recuperare dati dall'archiviazione remota.	N/D	max (10, v CPUs * 0.67) dove v CPUs dipende dalla dimensione dell'ista nza del broker

Nome	Descrizione	Valore predefinito per il cluster con l'archivi azione non a più livelli	Valore predefinito per il cluster abilitato all'archiviazione a più livelli
remote.storage.ena ble	Abilita l'archiviazione a più livelli (remota) per un argomento, se impostato su true. Disabilita l'archivi azione a più livelli a livello di argomento se impostato su false e remote.lo g.msk.disable.policy è impostato su Delete. Quando si disabilita l'archiviazione a più livelli, si eliminano i dati dall'archiviazione remota. Una volta disabilitata l'archivi azione a più livelli su un argomento, non sarà possibile riabilita rla.	false	false

Nome	Descrizione	Valore predefinito per il cluster con l'archiviazione non a più livelli	Valore predefinito per il cluster abilitato all'archiviazione a più livelli
replica.lag.time.max.ms	Se un follower non ha inviato richieste di fetch o non ha consumato fino all'offset di fine log del leader per almeno questo numero di millisecondi, il leader rimuove il follower dall'ISR.	30000	30000

Nome	Descrizione	Valore predefinito per il cluster con l'archivi azione non a più livelli	Valore predefinito per il cluster abilitato all'archiviazione a più livelli
retention.ms	Campo obbligatorio. Il tempo minimo è 3 giorni. Non esiste un'impostazione predefinita perché l'impostazione è obbligatoria. Amazon MSK utilizza il valore retention.ms con local.retention.ms per determinare quando i dati vengono spostati dall'arch iviazione locale a quella a più livelli. Il valore local.ret ention.ms specifica quando spostare i dati dall'archiviazione locale a quella a più livelli. Il valore retention.ms specifica quando rimuovere i dati dallo storage su più livelli (ovvero, rimossi dal cluster). Valori validi: numeri interi in [-1; +Inf]	Minimo 259.200.000 millisecondi (3 giorni). -1 per una conservaz ione infinita.	Minimo 259.200.000 millisecondi (3 giorni). -1 per una conservaz ione infinita.

Nome	Descrizione	Valore predefinito per il cluster con l'archivi azione non a più livelli	Valore predefinito per il cluster abilitato all'archiviazione a più livelli
socket.receive.buf fer.bytes	Buffer SO_RCVBUF dei socket del server dei socket. Se il valore è -1, viene utilizzato il sistema operativo predefinito.	102400	102400
socket.request.max .bytes	Numero massimo di byte in una richiesta socket.	104857600	104857600
socket.send.buffer .bytes	Buffer SO_SNDBUF dei socket del server dei socket. Se il valore è -1, viene utilizzato il sistema operativo predefinito.	102400	102400
unclean.leader.ele ction.enable	Indica se desideri che le repliche non incluse nel set ISR fungano da leader come ultima risorsa, anche se ciò potrebbe comportare la perdita di dati.	true	false
zookeeper.session. timeout.ms	Il timeout della sessione Apache in millisecondi. ZooKeeper	18000	18000
zookeeper.set.acl	Il client impostato da usare secure. ACLs	false	false

Per informazioni su come specificare valori di configurazione personalizzati, consulta la pagina the section called “Configurazioni Amazon MSK personalizzate”.

Linee guida per la configurazione a livello di argomento dello storage su più livelli di Amazon MSK

Di seguito sono riportate le impostazioni e le limitazioni predefinite per la configurazione dell'archiviazione a più livelli a livello di argomento.

- Amazon MSK non supporta segmenti di log di dimensioni inferiori per argomenti con l'archiviazione a più livelli attivata. Se si desidera creare un segmento, è prevista una dimensione minima del segmento di log di 48 MiB o un tempo minimo di distribuzione del segmento di 10 minuti. Questi valori sono mappati alle proprietà segment.bytes e segment.ms.
- Il valore di local.retention. ms/bytes can't equal or exceed the retention.ms/bytes. Questa è l'impostazione di conservazione dell'archiviazione a più livelli.
- Il valore predefinito per local.retention. ms/bytes is -2. This means that the retention.ms value is used for local.retention.ms/bytes. In questo caso, i dati rimangono sia nell'archiviazione locale sia nell'archiviazione a più livelli (una copia per ciascuna) e scadono insieme. Con questa opzione, una copia dei dati locali viene memorizzata nell'archiviazione remota. In questo caso, i dati letti dal traffico di utilizzo provengono dall'archiviazione locale.
- Il valore predefinito per retention.ms è 7 giorni. Non esiste un limite di dimensione predefinito per retention.bytes.
- Il valore minimo per retention.ms/bytes è -1. Ciò significa una conservazione infinita.
- Il valore minimo per local.retention. ms/bytes is -2. This means infinite retention for local storage. It matches with the retention.ms/bytesimpostazione come -1.
- La configurazione a livello di argomento retention.ms è obbligatoria per gli argomenti con lo storage su più livelli attivato. Il valore minimo per retention.ms è 3 giorni.

Per ulteriori informazioni sui vincoli di storage su più livelli, vedere. Vincoli e limitazioni dello storage su più livelli per i cluster Amazon MSK

Configurazioni del broker Express

Apache Kafka dispone di centinaia di configurazioni di broker che puoi utilizzare per ottimizzare le prestazioni del tuo cluster MSK Provisioned. L'impostazione di valori errati o non ottimali può influire sull'affidabilità e sulle prestazioni del cluster. I broker Express migliorano la disponibilità e la durata dei cluster MSK Provisioned impostando valori ottimali per le configurazioni critiche e proteggendole dai comuni errori di configurazione. Esistono tre categorie di configurazioni basate sull'accesso

in lettura e scrittura: configurazioni di lettura/scrittura (modificabili), di sola lettura e non di lettura/scrittura. Alcune configurazioni utilizzano ancora il valore predefinito di Apache Kafka per la versione di Apache Kafka in esecuzione sul cluster. Le contrassegniamo come Apache Kafka Default.

Argomenti

- Configurazioni personalizzate del broker MSK Express (accesso in lettura/scrittura)
- Configurazioni di sola lettura di Express Brokers

Configurazioni personalizzate del broker MSK Express (accesso in lettura/scrittura)

Puoi aggiornare le configurazioni dei read/write broker utilizzando la funzionalità di aggiornamento della configurazione di Amazon MSK o l'API di Apache Kafka. AlterConfig Le configurazioni del broker Apache Kafka sono statiche o dinamiche. Le configurazioni statiche richiedono il riavvio del broker per poter applicare la configurazione, mentre le configurazioni dinamiche non richiedono il riavvio del broker. Per ulteriori informazioni sulle proprietà di configurazione e sulle modalità di aggiornamento, vedere Aggiornamento delle configurazioni del broker.

Argomenti

- Configurazioni statiche sui broker MSK Express
- Configurazioni dinamiche su Express Brokers
- Configurazioni a livello di argomento su Express Brokers

Configurazioni statiche sui broker MSK Express

Puoi usare Amazon MSK per creare un file di configurazione MSK personalizzato per impostare le seguenti proprietà statiche. Amazon MSK imposta e gestisce tutte le altre proprietà che non hai impostato. È possibile creare e aggiornare file di configurazione statici dalla console MSK o utilizzando il comando configurations.

Proprietà	Descrizione	Valore predefinito
allow.everyone.if.no.acl.found	Se vuoi impostare questa proprietà su false, assicurat i innanzitutto di definire Apache Kafka ACLs per il tuo cluster. Se impostate questa	true

Proprietà	Descrizione	Valore predefinito
	proprietà su false e non definite prima Apache Kafka ACLs, perderete l'accesso al cluster. In tal caso, puoi aggiornare nuovamente la configurazione e impostare questa proprietà su true per riottenere l'accesso al cluster.	
auto.create.topics.enable	Abilita la creazione automatica di un argomento sul server.	false
compression.type	Specificate il tipo di compressione finale per un determinato argomento. Questa configurazione accetta i codec di compressione standard: gzip, snappy, lz4, zstd. Questa configurazione accetta inoltreuncompressed , il che equivale a non comprimere e nullaproducer, e ciò significa mantenere il codec di compressione originale impostato dal produttore.	Apache Kafka (impostazione predefinita)
connections.max.idle.ms	Timeout delle connessioni inattive in millisecondi. I thread del processore dei socket del server chiudono le connessioni inattive per un periodo superiore al valore impostato per questa proprietà.	Apache Kafka predefinito

Proprietà	Descrizione	Valore predefinito
delete.topic.enable	Abilita l'operazione di eliminazione argomento. Se questa configurazione è disattivata, non è possibile eliminare un argomento tramite lo strumento di amministrazione.	Apache Kafka predefinito
group.initial.rebalance.delay.ms	Il tempo durante il quale il coordinatore del gruppo attende che altri consumatori si uniscano a un nuovo gruppo prima di eseguire il primo ribilanciamento. Un ritardo più lungo significa potenzialmente meno ribilanciamenti, ma aumenta il tempo prima dell'inizio dell'elaborazione.	Apache Kafka predefinito
group.max.session.timeout.ms	Timeout sessione massimo per i consumatori registrati. Timeout più lunghi offrono ai consumatori più tempo per elaborare i messaggi tra heartbeat, ma implicano un aumento del tempo richiesto per rilevare gli errori.	Apache Kafka predefinito
leader.imbalance.per.broker.percentage	Il rapporto di squilibrio leader consentito per broker. Il controller attiva un bilanciamento dei leader se supera questo valore per broker. Questo valore è specificato in percentuale.	Apache Kafka predefinito

Proprietà	Descrizione	Valore predefinito
log.cleanup.policy	La policy di pulizia predefinita per i segmenti oltre la finestra di conservazione. Un elenco separato da virgole di policy valide. Policy valide sono `delete` e `compact`. Per i cluster abilitati allo storage su più livelli, è valida solo la policy. `delete`	Apache Kafka (impostazione predefinita)
log.message.timestamp.after.max.ms	La differenza di timestamp consentita tra il timestamp del messaggio e il timestamp del broker. Il timestamp del messaggio può essere successivo o uguale al timestamp del broker, con la differenza massima consentita determinata dal valore impostato in questa configurazione. Se `log.message.timestamp.type=CreateTime`, il messaggio verrà rifiutato se la differenza nei timestamp supera la soglia specificata. Questa configurazione viene ignorata se. `log.message.timestamp.type=LogAppendTime`	86400000 (24 * 60 * 60 * 1000 ms, ovvero 1 giorno)

Proprietà	Descrizione	Valore predefinito
log.message.timestamp.before.max.ms	La differenza di timestamp consentita tra il timestamp del broker e il timestamp del messaggio. Il timestamp del messaggio può essere precedente o uguale al timestamp del broker, con la differenza massima consentita determinata dal valore impostato in questa configurazione. Se `log.message.timestamp.type=CreateTime`, il messaggio verrà rifiutato se la differenza nei timestamp supera la soglia specificata. Questa configurazione viene ignorata se `log.message.timestamp.type=LogAppendTime`	86400000 (24 * 60 * 60 * 1000 ms, ovvero 1 giorno)
log.message.timestamp.type	Specifica se il timestamp nel messaggio è l'ora di creazione del messaggio o l'ora di aggiunta del log. I valori consentiti sono `CreateTime` e `LogAppendTime`.	Apache Kafka predefinito
log.retention.bytes	Dimensione massima del log prima dell'eliminazione.	Apache Kafka predefinito
log.retention.ms	Numero di millisecondi per conservare un file di registro prima di eliminarlo.	Apache Kafka (impostazione predefinita)

Proprietà	Descrizione	Valore predefinito
numero massimo di connessioni per ip	Il numero massimo di connessioni consentite da ogni indirizzo IP. Questo valore può essere impostato su `0` se sono presenti sostituzioni configurate utilizzando la `max.connections.per.ip.overrides` proprietà. Le nuove connessioni dall'indirizzo IP vengono interrotte se viene raggiunto il limite.	Apache Kafka (impostazione predefinita)
max.incremental.fetch.session.cache.slots	Numero massimo di sessioni di recupero incrementali che vengono mantenute.	Apache Kafka predefinito

Proprietà	Descrizione	Valore predefinito
message.max.bytes	Dimensione massima del batch di record consentit a da Kafka. Se aumenti questo valore e sono presenti consumatori più vecchi di 0.10.2, anche le dimensioni di recupero dei consumatori devono essere incrementate in modo che possano recuperar e batch di record di queste dimensioni. Nella versione più recente del formato del messaggio, i messaggi vengono sempre raggruppati in batch per maggiore efficienza. Nelle versioni precedenti del formato del messaggio, i record non compressi non sono raggruppati in batch; in tal caso, questo limite si applica solo a un singolo record. Puoi impostare questo valore per argomento con la configura zione a livello di argomento. `max.message.bytes`	Apache Kafka predefinito
num.partitions	Numero predefinito di partizion i per argomento.	1

Proprietà	Descrizione	Valore predefinito
offsets.retention.minutes	Dopo che un gruppo di consumatori perde tutti i suoi consumatori (ovvero, diventa vuoto) i suoi offset vengono mantenuti per questo periodo di conservazione prima di essere scartati. Per i consumatori autonomi (ovvero quelli che utilizzano l'assegnazione manuale), gli offset scadono dopo l'ultimo commit più questo periodo di conservazione.	Apache Kafka predefinito
replica.fetch.max.bytes	Numero di byte di messaggi da recuperare per ogni partizione. Questo valore non è un massimo assoluto. Se il primo batch di record nella prima partizione non vuota del recupero è più grande di questo valore, viene restituito il batch di record per garantire l'avanzamento. La proprietà message.max.bytes (configurazione broker) o max.message.bytes (configur azione argomento) specifica la dimensione massima del batch di record accettata dal broker.	Apache Kafka predefinito

Proprietà	Descrizione	Valore predefinito
replica.selector.class	Il nome completo della classe che implementa. ReplicaSelector Il broker utilizza questo valore per trovare la replica di lettura preferita. Se desideri consentire ai consumatori di eseguire il recupero dalla replica più vicina, imposta questa proprietà su. `org.apache.kafka.common.replica.RackAwareReplicaSelector`	Apache Kafka (impostazione predefinita)
socket.receive.buffer.bytes	Buffer SO_RCVBUF dei socket del server dei socket. Se il valore è -1, viene utilizzato il sistema operativo predefinito.	102400
socket.request.max.bytes	Numero massimo di byte in una richiesta socket.	104857600
socket.send.buffer.bytes	Buffer SO_SNDBUF dei socket del server dei socket. Se il valore è -1, viene utilizzato il sistema operativo predefinito.	102400

Proprietà	Descrizione	Valore predefinito
transaction.max.timeout.ms	Timeout massimo per transazioni. Se il tempo di transazione richiesto da un cliente supera questo valore, il broker restituisce un errore in. InitProducerIdRequest Ciò evita un timeout troppo elevato per un client, che può rallentar e i consumatori che leggono dagli argomenti inclusi nella transazione.	Apache Kafka (impostazione predefinita)

Proprietà	Descrizione	Valore predefinito
transactional.id.expiration.ms	Il tempo in millisecondi durante il quale il coordinatore della transazione attende di ricevere eventuali aggiornamenti sullo stato delle transazioni per la transazione corrente prima che il coordinatore faccia scadere il proprio ID transazionale. Questa impostazione influenza anche la scadenza dell'ID del produttore perché fa scadere IDs il produttore quando questo tempo è trascorso dall'ultima scrittura con l'ID produttore specificato. Producer IDs potrebbe scadere prima se l'ultima scrittura dall'ID produttore viene eliminata a causa delle impostazioni di conservazione dell'argomento. Il valore minimo per questa proprietà è 1 millisecondo.	Apache Kafka (impostazione predefinita)

Configurazioni dinamiche su Express Brokers

Puoi utilizzare l' AlterConfig API Apache Kafka o lo strumento Kafka-configs.sh per modificare le seguenti configurazioni dinamiche. Amazon MSK imposta e gestisce tutte le altre proprietà che non hai impostato. Puoi impostare dinamicamente proprietà di configurazione a livello di cluster e di broker che non richiedono il riavvio del broker.

Proprietà	Descrizione	Valore predefinito
advertised.listeners	Listener da pubblicare per l'utilizzo da parte dei client, se diversi dalla proprietà config. `listeners` Negli ambienti IaaS, potrebbe essere necessario che questa sia diversa dall'interfaccia a cui si collega il broker. Se questo non è impostato, verrà utilizzato il valore per gli ascoltatori. A differenza degli ascoltatori, non è valido pubblicizzare il meta-indirizzo 0.0.0.0. Inoltre`listener`, a differenza di questa proprietà, possono	null

Proprietà	Descrizione	Valore predefinito
	esserci porte duplicate, in modo che un listener possa essere configurato per annunciare l'indirizzo di un altro listener. Ciò può essere utile in alcuni casi in cui vengono utilizzati sistemi di bilanciam ento del carico esterni. Questa proprietà è impostata a livello di broker.	

Proprietà	Descrizione	Valore predefinito
compressi on.type	Il tipo di compressione finale per un determinato argomento. Puoi impostare questa proprietà sui codec di compressi one standard (`gzip`, `snappy`, `lz4` e `zstd`). Inoltre, accetta `uncompres sed` . Questo valore equivale a nessuna compressione. Se imposti il valore su `producer`, significa mantenere il codec di compressi one originale impostato dal produttore.	Apache Kafka (impostaz ione predefinita)

Proprietà	Descrizione	Valore predefinito
log.cleaner.delete.retention.ms	Il periodo di tempo necessario per conservare i marker di eliminazione di tombstone per gli argomenti con log compattati. Questa impostazione stabilisce anche un limite al tempo in cui un consumatore deve completare una lettura se inizia dall'offset 0 per assicurarsi di ottenere un'istantanea valida della fase finale. Altrimenti, le lapidi eliminate potrebbero essere raccolte prima che completino la scansione.	86400000 (24 * 60 * 60 * 1000 ms, ovvero 1 giorno), Apache Kafka Default

Proprietà	Descrizione	Valore predefinito
log.cleaner.min.compaction.lag.ms	Il periodo minimo di tempo in cui un messaggio rimarrà non compattato nel registro. Questa impostazione è applicabile solo ai log che vengono compattati.	0, Apache Kafka predefinito

Proprietà	Descrizione	Valore predefinito
log.cleaner.max.compaction.lag.ms	Il periodo massimo di tempo in cui un messaggio rimarrà non idoneo per la compattazione nel registro. Questa impostazione è applicabile solo ai log che vengono compattati. Questa configurazione sarebbe limitata all'intervallo di [7 giorni, Long.Max].	9223372036854775807, impostazione predefinita di Apache Kafka

Proprietà	Descrizione	Valore predefinito
log.clean up.policy	La policy di pulizia predefinita per i segmenti oltre la finestra di conservaz ione. Un elenco separato da virgole di policy valide. Policy valide sono `delete` e `compact`. Per i cluster abilitati allo storage su più livelli, è valida solo la policy. `delete`	Apache Kafka (impostaz ione predefinita)

Proprietà	Descrizione	Valore predefinito
log.message.timestamp.after.max.ms	La differenza di timestamp consentita tra il timestamp del messaggio e il timestamp del broker. Il timestamp del messaggio può essere successivo o uguale al timestamp del broker, con la differenza massima consentita determinata dal valore impostato in questa configurazione. Se `log.message.timestamp.type=CreateTime` , il messaggio verrà rifiutato se la differenza nei timestamp supera la soglia specificata. Questa	86400000 (24 * 60 * 60 * 1000 ms, ovvero 1 giorno)

Proprietà	Descrizione	Valore predefinito
	configurazione viene ignorata se. `log.message.timestamp.type=LogAppendTime`	

Proprietà	Descrizione	Valore predefinito
log.message.timestamp.before.max.ms	La differenza di timestamp consentita tra il timestamp del broker e il timestamp del messaggio. Il timestamp del messaggio può essere precedente o uguale al timestamp del broker, con la differenza massima consentita determinata dal valore impostato in questa configurazione. Se `log.message.timestamp.type=CreateTime`, il messaggio verrà rifiutato se la differenza nei timestamp supera la soglia specificata. Questa	86400000 (24 * 60 * 60 * 1000 ms, ovvero 1 giorno)

Proprietà	Descrizione	Valore predefinito
	configurazione viene ignorata se. `log.message.timestamp.type=LogAppendTime`	
log.message.timestamp.type	Specifica se il timestamp nel messaggio è l'ora di creazione del messaggio o l'ora di aggiunta del log. I valori consentiti sono `CreateTime` e `LogAppendTime` .	Apache Kafka predefinito
log.retention.bytes	Dimensione massima del log prima dell'eliminazione.	Apache Kafka predefinito
log.retention.ms	Numero di millisecondi per conservare un file di registro prima di eliminarlo.	Apache Kafka (impostazione predefinita)

Proprietà	Descrizione	Valore predefinito
max.connection.creation.rate	La velocità massima di creazione della connessione consentita nel broker in qualsiasi momento.	Apache Kafka (impostazione predefinita)
numero massimo di connessioni	Il numero massimo di connessioni consentite nel broker in qualsiasi momento. Questo limite viene applicato in aggiunta a qualsiasi limite per IP configurato utilizzando. `max.connections.per.ip`	Apache Kafka (impostazione predefinita)

Proprietà	Descrizione	Valore predefinito
numero massimo di connessioni per ip	Il numero massimo di connessioni consentite da ogni indirizzo IP. Questo valore può essere impostato su `0` se sono presenti sostituzioni configurate utilizzando la proprietà max.connections.per.ip.overrides. Le nuove connessioni dall'indirizzo IP vengono interrotte se viene raggiunto il limite.	Apache Kafka (impostazione predefinita)

Proprietà	Descrizione	Valore predefinito
max.connections.per.ip.overrides	Un elenco separato da virgole di per-ip o nome host sostituisce il numero massimo predefinito di connessioni. Un valore di esempio è `hostName:100,127.0.0.1:200`	Apache Kafka predefinito

Proprietà	Descrizione	Valore predefinito
message.max.bytes	Dimensione massima del batch di record consentita da Kafka. Se aumenti questo valore e sono presenti consumatori più vecchi di 0.10.2, anche le dimensioni di recupero dei consumatori devono essere incrementate in modo che possano recuperare batch di record di queste dimensioni. Nella versione più recente del formato del messaggio, i messaggi vengono sempre raggruppati in batch per maggiore efficienza. Nelle versioni	Apache Kafka predefinito

Proprietà	Descrizione	Valore predefinito
	precedenti del formato del messaggio, i record non compressi non sono raggruppati in batch; in tal caso, questo limite si applica solo a un singolo record. Puoi impostare questo valore per argomento con la configurazione a livello di argomento `.max.message.bytes`	

Proprietà	Descrizione	Valore predefinito
producer.id.expiration.ms	Il tempo in ms che il leader della partizione di un argomento aspetterà prima che il produttore scada. IDs Il produttore non IDs scadrà finché una transazione ad esso associata è ancora in corso. Tieni presente che producer IDs potrebbe scadere prima se l'ultima scrittura dall'ID del produttore viene eliminata a causa delle impostazioni di conservazione dell'argomento. L'impostazione di questo valore uguale o superiore a `delivery.timeout.ms` può aiutare	Apache Kafka (impostazione predefinita)

Proprietà	Descrizione	Valore predefinito
	a prevenire la scadenza durante i nuovi tentativi e a proteggere dalla duplicazione dei messaggi, ma l'impostazione predefinita dovrebbe essere ragionevole per la maggior parte dei casi d'uso.	

Configurazioni a livello di argomento su Express Brokers

Puoi utilizzare i comandi Apache Kafka per impostare o modificare le proprietà di configurazione a livello dell'argomento per argomenti nuovi ed esistenti. Se non puoi fornire alcuna configurazione a livello di argomento, Amazon MSK utilizza il broker predefinito. Come per le configurazioni a livello di broker, Amazon MSK protegge alcune proprietà di configurazione a livello di argomento da eventuali modifiche. Gli esempi includono il fattore di replica e. `min.insync.replicas` `unclean.leader.election.enable` Se tenti di creare un argomento con un valore del fattore di replica diverso da`3`, Amazon MSK creerà l'argomento con un fattore di replica di `3` default. Per ulteriori informazioni sulle proprietà di configurazione a livello di argomento ed esempi di come impostarle, consulta la pagina Topic-Level Configs nella documentazione ufficiale di Apache Kafka.

Proprietà	Descrizione
cleanup.policy	Questa configurazione indica la politica di conservazione da utilizzare sui segmenti di log. La politica di «eliminazione» (che è l'impostazione predefinita) eliminerà i vecchi segmenti

Proprietà	Descrizione
	una volta raggiunto il tempo di conservazione o il limite di dimensione. La politica «compatta » consentirà la compattazione dei log, che conserva il valore più recente per ogni chiave. È anche possibile specificare entrambe le politiche in un elenco separato da virgole (ad esempio, «delete, compact»). In questo caso, i vecchi segmenti verranno eliminati in base alla configurazione del tempo di conservazione e delle dimensioni, mentre i segmenti mantenuti verranno compattati. La compattazione sui broker Express viene attivata dopo che i dati in una partizione raggiungono i 256 MB.
compression.type	Specificate il tipo di compressione finale per un determinato argomento. Questa configura zione accetta i codec di compressione standard (`gzip`,, `snappylz4`,`zstd`). Accetta inoltre ciò `uncompressed` che equivale a nessuna compressione; il `producer` che significa mantenere il codec di compressione originale impostato dal produttore.

Proprietà	Descrizione
delete.retention.ms	Il periodo di tempo necessario per conservar e i marker di eliminazione di tombstone per gli argomenti con log compattati. Questa impostazi one stabilisce anche un limite al tempo in cui un consumatore deve completare una lettura se inizia dall'offset 0 per assicurarsi di ottenere un'istantanea valida della fase finale. Altriment i, le lapidi eliminate potrebbero essere raccolte prima che completino la scansione. Il valore predefinito per questa impostazione è 86400000 (24 * 60 * 60 * 1000 ms, ovvero 1 giorno), Apache Kafka Default
max.message.bytes	La dimensione massima del batch di record consentita da Kafka (dopo la compressione, se la compressione è abilitata). Se questa cifra aumenta e ci sono consumatori più vecchi di età`0.10.2`, è necessario aumentare anche la dimensione di recupero dei consumatori in modo che possano recuperare batch di dischi di dimensioni così grandi. Nella versione più recente del tipo di formato, i record vengono sempre raggruppati in batch ai fini dell'effi cienza. Nelle versioni precedenti del tipo di formato, i record non compressi non sono raggruppati in batch e questo limite si applica solo a un singolo record in quel caso. Questo può essere impostato per argomento con il livello dell'argomento. `max.message.bytes config`

Proprietà	Descrizione
messaggio.timestamp.after.max.ms	Questa configurazione imposta la differenz a di timestamp consentita tra il timestamp del messaggio e il timestamp del broker. Il timestamp del messaggio può essere successiv o o uguale al timestamp del broker, con la differenza massima consentita determinata dal valore impostato in questa configurazione. Se`message.timestamp.type=Crea teTime` , il messaggio verrà rifiutato se la differenza nei timestamp supera la soglia specificata. Questa configurazione viene ignorata se. `message.timestamp. type=LogAppendTime`
message.timestamp.before.max.ms	Questa configurazione imposta la differenz a di timestamp consentita tra il timestamp del broker e il timestamp del messaggio . Il timestamp del messaggio può essere precedente o uguale al timestamp del broker, con la differenza massima consentit a determinata dal valore impostato in questa configurazione. Se`message.timestamp. type=CreateTime` , il messaggio verrà rifiutato se la differenza nei timestamp supera la soglia specificata. Questa configurazione viene ignorata se. `message.timestamp. type=LogAppendTime`
message.timestamp.type	Definisce se il timestamp nel messaggio è l'ora di creazione del messaggio o l'ora di aggiunta del registro. Il valore deve essere o `CreateTime` `LogAppendTime`

Proprietà	Descrizione
min.compaction.lag.ms	Il periodo minimo di tempo in cui un messaggio rimarrà non compattato nel registro. Questa impostazione è applicabile solo ai log che vengono compattati. Il valore predefinito per questa impostazione è 0, Apache Kafka Default
max.compaction.lag.ms	Il periodo massimo di tempo in cui un messaggio rimarrà non idoneo per la compattazione nel registro. Questa impostazione è applicabile solo ai log che vengono compattati. Questa configurazione sarebbe limitata all'intervallo di [7 giorni, Long.Max]. Il valore predefinito per questa impostazione è 9223372036854775807, Apache Kafka Default.
retention.bytes	Questa configurazione controlla la dimensione massima che una partizione (composta da segmenti di log) può raggiungere prima di eliminare i vecchi segmenti di registro per liberare spazio se utilizziamo la politica di conservazione «elimina». Per impostazione predefinita, non esiste un limite di dimensione, ma solo un limite di tempo. Poiché questo limite viene applicato a livello di partizione, moltiplicalo per il numero di partizioni per calcolare la conservazione dell'argomento in byte. Inoltre, funziona indipendentemente dalle configurazioni e dalle `retention.bytes configuration` configurazioni. `segment.ms segment.bytes` Inoltre, attiva il lancio di un nuovo segmento se `retention.bytes` è configurato a zero.

Proprietà	Descrizione
retention.ms	Questa configurazione controlla il tempo massimo di conservazione di un registro prima di eliminare i vecchi segmenti di registro per liberare spazio se utilizziamo la politica di conservazione «elimina». Ciò rappresenta uno SLA sulla tempistica con cui i consumatori devono leggere i propri dati. Se impostato su `-1`, non viene applicato alcun limite di tempo. Inoltre, la `retention.ms` configurazione funziona indipendentemente dalle `segment.bytes` configurazioni `segment.ms` e. Inoltre, attiva il lancio di un nuovo segmento se la `retention.ms` condizione è soddisfatta.

Configurazioni di sola lettura di Express Brokers

Amazon MSK imposta i valori per queste configurazioni e le protegge da modifiche che potrebbero influire sulla disponibilità del cluster. Questi valori possono cambiare a seconda della versione di Apache Kafka in esecuzione sul cluster, quindi ricordati di controllare i valori del tuo cluster specifico. Ecco alcuni esempi.

Configurazioni di sola lettura di Express Brokers

Proprietà	Descrizione	Valore di Express Broker
broker.id	L'id del broker per questo server.	1,2,3...
broker.rack	Rack del broker. Questo verrà utilizzato nell'assegnazione della replica in base al rack per la tolleranza ai guasti. Esempi: ``, RACK1 `us-east-1d`	ID AZ o ID di sottorete

Proprietà	Descrizione	Valore di Express Broker
default.replication.factor	Fattori di replica predefiniti per tutti gli argomenti.	3
fetch.max.bytes	Il numero massimo di byte che restituiremo per una richiesta di recupero.	Apache Kafka (impostazione predefinita)
dimensione massima del gruppo	Il numero massimo di consumatori che un singolo gruppo di consumatori può ospitare.	Apache Kafka (impostazione predefinita)
inter.broker.listener.name	Nome dell'ascoltatore utilizzato per la comunicazione tra i broker.	REPLICATION_SECURE o REPLICATION
inter.broker.protocol. version	Speciifica quale versione del protocollo inter-broker viene utilizzata.	Apache Kafka (impostazione predefinita)
ascoltatori	Elenco degli ascoltatori: elenco separato da virgole di URIs we will listening e dei nomi degli ascoltatori. È possibile impostare il `advertised.listeners property` , ma non la proprietà. `listeners`	Generato da MSK
log.message.format.version	Specificare la versione del formato del messaggio che il broker utilizzerà per aggiungere messaggi ai log.	Apache Kafka (impostazione predefinita)

Proprietà	Descrizione	Valore di Express Broker
min.insync.replicas	Quando un produttore imposta acks su `all` (or`-1`), il valore in `min.insync.replicas` specifica il numero minimo di repliche che devono confermare una scrittura affinché la scrittura sia considerata riuscita. Se questo minimo non può essere raggiunto, il produttore solleva un'eccezione (o`NotEnough Replicas`). `NotEnough ReplicasAfterAppend` Puoi utilizzare il valore degli ack del tuo produttore per far rispettare maggiori garanzie di durabilità. Impostando acks su «all». Ciò garantisce che il produttore generi un'eccezione se la maggior parte delle repliche non riceve una scrittura.	2
num.io.thread	Numero di thread utilizzati dal server per produrre richieste , che possono includere l'I/O del disco. (m7g.large, 8), (m7g.xlarge, 8), (m7g.2xlarge, 16), (m7g.4xlarge, 32), (m7g.8xlarge, 64), (m7g.12xlarge, 96), (m7g.16xlarge, 128)	In base al tipo di istanza. =Math.max (8, 2* v) CPUs

Proprietà	Descrizione	Valore di Express Broker
num.network.threads	Numero di thread utilizzati dal server per ricevere richieste dalla rete e inviare risposte alla rete. (m7g.large, 8), (m7g.xlarge, 8), (m7g.2xla rge, 8), (m7g.4xlarge, 16), (m7g.8xlarge, 32), (m7g.12xl arge, 48), (m7g.16xlarge, 64)	In base al tipo di istanza. =Math.max (8, v) CPUs
replica.fetch.response.max. bytes	Il numero massimo di byte previsto per l'intera risposta di recupero. I record vengono recuperati in batch. Se il primo batch di record nella prima partizione non vuota del recupero è più grande di questo valore, il batch di record verrà comunque restituito per garantire l'avanzamento. Questo non è un massimo assoluto. Le proprietà `message.m ax.bytes` (broker config) o `max.message.bytes` (topic config) specificano la dimensione massima del batch di record che il broker accetta.	Apache Kafka (impostazione predefinita)

Proprietà	Descrizione	Valore di Express Broker
request.timeout.ms	La configurazione controlla il tempo massimo di attesa del client per la risposta di una richiesta. Se la risposta non viene ricevuta prima dello scadere del timeout, il client invierà nuovamente la richiesta se necessario o fallirà la richiesta se i nuovi tentativi sono esauriti.	Apache Kafka predefinito
transaction.state.log.min.isr	`min.insync.replicas` Configurazione sostituita per l'argomento della transazione.	2
transaction.state.log.replication.factor	Il fattore di replica per l'argomento di transazione.	Apache Kafka (impostazione predefinita)
unclean.leader.election.enable	Consente alle repliche non incluse nel set ISR di fungere da leader come ultima risorsa, anche se ciò potrebbe comportare la perdita di dati.	FALSE

Operazioni di configurazione del broker

Le configurazioni del broker Apache Kafka sono statiche o dinamiche. Le configurazioni statiche richiedono il riavvio del broker per poter applicare la configurazione. Le configurazioni dinamiche non richiedono il riavvio del broker per aggiornare la configurazione. Per ulteriori informazioni sulle proprietà di configurazione e sulle modalità di aggiornamento, consulta Configurazione di Apache Kafka.

In questo argomento viene descritto come creare configurazioni MSK personalizzate e come eseguire operazioni su di esse. Per informazioni su come utilizzare configurazioni MSK per creare o aggiornare cluster, consulta the section called “Caratteristiche e concetti chiave”.

Argomenti

Creazione di una configurazione

Questo processo descrive come creare una configurazione Amazon MSK personalizzata e come eseguire operazioni su di essa.

1. Creare un file in cui specificare le proprietà di configurazione che si desidera impostare e i valori da assegnare alle stesse. Di seguito sono riportati i contenuti di un file di configurazione di esempio.

```
auto.create.topics.enable = true

log.roll.ms = 604800000
```

2. Esegui il AWS CLI comando seguente e sostituiscilo *`config-file-path`* con il percorso del file in cui hai salvato la configurazione nel passaggio precedente.

> **Note**
>
> Il nome scelto per la configurazione deve corrispondere alla seguente espressione regolare: "^[0-9A-Za-z][0-9A-Za-z-]{0,}$".

```
aws kafka create-configuration --name "ExampleConfigurationName" --description
 "Example configuration description." --kafka-versions "1.1.1" --server-properties
 fileb://config-file-path
```

Di seguito è riportato un esempio di una risposta corretta dopo l'esecuzione di questo comando.

```
{
    "Arn": "arn:aws:kafka:us-east-1:123456789012:configuration/SomeTest/
abcdabcd-1234-abcd-1234-abcd123e8e8e-1",
    "CreationTime": "2019-05-21T19:37:40.626Z",
    "LatestRevision": {
        "CreationTime": "2019-05-21T19:37:40.626Z",
        "Description": "Example configuration description.",
        "Revision": 1
    },
    "Name": "ExampleConfigurationName"
}
```

3. Il comando precedente restituisce un nome della risorsa Amazon (ARN) per la configurazione appena creata. Salvare questo ARN perché occorre per fare riferimento a questa configurazione in altri comandi. Se l'ARN della configurazione viene perso, è possibile trovarlo nuovamente elencando tutte le configurazioni presenti nell'account.

Aggiornare la configurazione

Questo processo descrive come aggiornare una configurazione Amazon MSK personalizzata.

1. Crea un file in cui specificare le proprietà di configurazione che desideri aggiornare e i valori da assegnare alle stesse. Di seguito sono riportati i contenuti di un file di configurazione di esempio.

```
auto.create.topics.enable = true

min.insync.replicas = 2
```

2. Esegui il AWS CLI comando seguente e sostituiscilo *config-file-path* con il percorso del file in cui hai salvato la configurazione nel passaggio precedente.

   Sostituisci *configuration-arn* con l'ARN che hai ottenuto quando hai creato la configurazione. Se l'ARN non è stato salvato al momento della creazione della configurazione,

è possibile utilizzare il comando `list-configurations` per elencare tutte le configurazioni presenti nell'account. La configurazione desiderata viene visualizzata nell'elenco di risposta. L'ARN della configurazione viene visualizzato anche in tale elenco.

```
aws kafka update-configuration --arn configuration-arn --description "Example
 configuration revision description." --server-properties fileb://config-file-path
```

3. Di seguito è riportato un esempio di una risposta corretta dopo l'esecuzione di questo comando.

```
{
    "Arn": "arn:aws:kafka:us-east-1:123456789012:configuration/SomeTest/
abcdabcd-1234-abcd-1234-abcd123e8e8e-1",
    "LatestRevision": {
        "CreationTime": "2020-08-27T19:37:40.626Z",
        "Description": "Example configuration revision description.",
        "Revision": 2
    }
}
```

Eliminare la configurazione

Nella procedura seguente viene illustrato come eliminare una configurazione non collegata a un cluster. Non è possibile eliminare una configurazione collegata a un cluster.

1. Per eseguire questo esempio, sostituiscilo *configuration-arn* con l'ARN che hai ottenuto quando hai creato la configurazione. Se l'ARN non è stato salvato al momento della creazione della configurazione, è possibile utilizzare il comando `list-configurations` per elencare tutte le configurazioni presenti nell'account. La configurazione desiderata viene visualizzata nell'elenco di risposta. L'ARN della configurazione viene visualizzato anche in tale elenco.

```
aws kafka delete-configuration --arn configuration-arn
```

2. Di seguito è riportato un esempio di una risposta corretta dopo l'esecuzione di questo comando.

```
{
    "arn": " arn:aws:kafka:us-east-1:123456789012:configuration/SomeTest/
abcdabcd-1234-abcd-1234-abcd123e8e8e-1",
    "state": "DELETING"
}
```

Ottieni i metadati di configurazione

La procedura seguente mostra come descrivere una configurazione Amazon MSK per ottenere i metadati relativi alla configurazione.

1. Il comando seguente restituisce i metadati relativi alla configurazione. Per ottenere una descrizione dettagliata della configurazione, eseguire `describe-configuration-revision`.

   Per eseguire questo esempio, sostituiscilo *`configuration-arn`* con l'ARN che hai ottenuto quando hai creato la configurazione. Se l'ARN non è stato salvato al momento della creazione della configurazione, è possibile utilizzare il comando `list-configurations` per elencare tutte le configurazioni presenti nell'account. La configurazione desiderata viene visualizzata nell'elenco di risposta. L'ARN della configurazione viene visualizzato anche in tale elenco.

   ```
   aws kafka describe-configuration --arn configuration-arn
   ```

2. Di seguito è riportato un esempio di una risposta corretta dopo l'esecuzione di questo comando.

   ```
   {
       "Arn": "arn:aws:kafka:us-east-1:123456789012:configuration/SomeTest/abcdabcd-
   abcd-1234-abcd-abcd123e8e8e-1",
       "CreationTime": "2019-05-21T00:54:23.591Z",
       "Description": "Example configuration description.",
       "KafkaVersions": [
           "1.1.1"
       ],
       "LatestRevision": {
           "CreationTime": "2019-05-21T00:54:23.591Z",
           "Description": "Example configuration description.",
           "Revision": 1
       },
       "Name": "SomeTest"
   }
   ```

Ottieni dettagli sulla revisione della configurazione

Questo processo consente di ottenere una descrizione dettagliata della revisione della configurazione di Amazon MSK.

Se utilizzi il comando `describe-configuration` per descrivere una configurazione MSK, visualizzerai i metadati della configurazione. Per ottenere una descrizione dettagliata della configurazione, utilizza il comando `describe-configuration-revision`.

- Esegui il comando seguente e sostituiscilo *`configuration-arn`* con l'ARN ottenuto quando hai creato la configurazione. Se l'ARN non è stato salvato al momento della creazione della configurazione, è possibile utilizzare il comando `list-configurations` per elencare tutte le configurazioni presenti nell'account. La configurazione desiderata viene visualizzata nell'elenco di risposta. L'ARN della configurazione viene visualizzato anche in tale elenco.

  ```
  aws kafka describe-configuration-revision --arn configuration-arn --revision 1
  ```

  Di seguito è riportato un esempio di una risposta corretta dopo l'esecuzione di questo comando.

  ```
  {
      "Arn": "arn:aws:kafka:us-east-1:123456789012:configuration/SomeTest/abcdabcd-
  abcd-1234-abcd-abcd123e8e8e-1",
      "CreationTime": "2019-05-21T00:54:23.591Z",
      "Description": "Example configuration description.",
      "Revision": 1,
      "ServerProperties":
   "YXV0by5jcmVhdGUudG9waWNzLmVuYWJsZSA9IHRydWUKCgp6b29rZWVwZXIuY29ubmVjdGlvbi50aW1lb3V0Lm1zI
  }
  ```

  Il valore di `ServerProperties` è codificato con base64. Se si utilizza un decodificatore base64 (ad esempio, https://www.base64decode.org/) per decodificarlo manualmente, si ottiene il contenuto del file di configurazione originale utilizzato per creare la configurazione personalizzata. In questo caso, si ottiene quanto segue:

  ```
  auto.create.topics.enable = true

  log.roll.ms = 604800000
  ```

Elenca le configurazioni presenti nel tuo account per la regione corrente

Questo processo descrive come elencare tutte le configurazioni Amazon MSK nel tuo account per la regione corrente AWS .

- Esegui il comando seguente.

```
aws kafka list-configurations
```

Di seguito è riportato un esempio di una risposta corretta dopo l'esecuzione di questo comando.

```
{
    "Configurations": [
        {
            "Arn": "arn:aws:kafka:us-east-1:123456789012:configuration/SomeTest/
abcdabcd-abcd-1234-abcd-abcd123e8e8e-1",
            "CreationTime": "2019-05-21T00:54:23.591Z",
            "Description": "Example configuration description.",
            "KafkaVersions": [
                "1.1.1"
            ],
            "LatestRevision": {
                "CreationTime": "2019-05-21T00:54:23.591Z",
                "Description": "Example configuration description.",
                "Revision": 1
            },
            "Name": "SomeTest"
        },
        {
            "Arn": "arn:aws:kafka:us-east-1:123456789012:configuration/SomeTest/
abcdabcd-1234-abcd-1234-abcd123e8e8e-1",
            "CreationTime": "2019-05-03T23:08:29.446Z",
            "Description": "Example configuration description.",
            "KafkaVersions": [
                "1.1.1"
            ],
            "LatestRevision": {
                "CreationTime": "2019-05-03T23:08:29.446Z",
                "Description": "Example configuration description.",
                "Revision": 1
            },
            "Name": "ExampleConfigurationName"
        }
    ]
}
```

Stati di configurazione di Amazon MSK

Una configurazione Amazon MSK può trovarsi in uno dei seguenti stati. Per eseguire un'operazione su una configurazione, la configurazione deve trovarsi nello stato `ACTIVE` o `DELETE_FAILED`:

- `ACTIVE`
- `DELETING`
- `DELETE_FAILED`

Rattoppare

Applicazione di patch sui cluster MSK Provisioned

Periodicamente, Amazon MSK aggiorna il software sui broker del tuo cluster. La manutenzione include aggiornamenti pianificati o riparazioni non pianificate. La manutenzione pianificata include aggiornamenti del sistema operativo, aggiornamenti di sicurezza e altri aggiornamenti software necessari per mantenere l'integrità, la sicurezza e le prestazioni del cluster. Eseguiamo una manutenzione non pianificata per risolvere il degrado improvviso dell'infrastruttura. Eseguiamo la manutenzione su broker Standard ed Express, ma le esperienze sono diverse.

Applicazione di patch per broker Standard

Gli aggiornamenti ai broker Standard non hanno alcun impatto sulle scritture e sulle letture delle applicazioni se segui le best practice.

Amazon MSK utilizza aggiornamenti periodici per il software per mantenere un'elevata disponibilità dei cluster. Durante questo processo, i broker vengono riavviati uno alla volta e Kafka trasferisce automaticamente la leadership a un altro broker online. I client Kafka dispongono di meccanismi integrati per rilevare automaticamente il cambio di leadership per le partizioni e continuare a scrivere e leggere dati in un cluster MSK. Segui le istruzioni Best practice per i client Apache Kafka per garantire il corretto funzionamento del cluster in ogni momento, anche durante l'applicazione delle patch.

In seguito alla disconnessione di un broker, è normale riscontrare errori di disconnessione transitori sui clienti. Inoltre, per una breve finestra (fino a 2 minuti, in genere meno), osserverete alcuni picchi nella latenza di lettura e scrittura di p99 (in genere alti millisecondi, fino a ~2 secondi). Questi picchi sono previsti e sono causati dalla riconnessione del client a un nuovo broker leader; non influiscono sulla produzione o sul consumo e si risolveranno dopo la riconnessione. Per ulteriori informazioni, consulta Broker offline e client failover.

Noterai anche un aumento della metrica`UnderReplicatedPartitions`, previsto poiché le partizioni del broker che è stato chiuso non replicano più i dati. Ciò non ha alcun impatto sulle scritture e le letture delle applicazioni, poiché le repliche di queste partizioni ospitate su altri broker ora soddisfano le richieste.

Dopo l'aggiornamento del software, quando il broker torna online, deve «recuperare il ritardo» sui messaggi prodotti mentre era offline. Durante il catch up, si può anche osservare un aumento dell'utilizzo del volume, del throughput e della CPU. Questi non dovrebbero avere alcun impatto sulle scritture e le letture nel cluster se i broker dispongono di risorse sufficienti di CPU, memoria, rete e volume.

Applicazione di patch per i broker Express

Non ci sono finestre di manutenzione per i broker Express. Amazon MSK aggiorna automaticamente il cluster su base continuativa in modo distribuito nel tempo, il che significa che puoi aspettarti riavvii occasionali e singolari del broker nel corso del mese. In questo modo non è necessario elaborare piani o adattamenti in base a finestre di manutenzione una tantum a livello di cluster. Come sempre, il traffico rimarrà ininterrotto durante il riavvio del broker poiché la leadership passerà ad altri broker che continueranno a soddisfare le richieste.

I broker Express sono configurati con impostazioni e protezioni basate sulle best practice che rendono il cluster resiliente alle modifiche di carico che possono verificarsi durante la manutenzione. Amazon MSK imposta quote di throughput sui broker Express per mitigare l'impatto del sovraccarico del cluster, che può causare problemi durante il riavvio del broker. Questi miglioramenti eliminano la necessità di notifiche anticipate, pianificazione e finestre di manutenzione quando si utilizzano i broker Express.

I broker Express replicano sempre i dati in tre modi, in modo che i client effettuino automaticamente il failover durante i riavvii. Non devi preoccuparti che gli argomenti diventino non disponibili a causa del fattore di replica impostato su 1 o 2. Inoltre, il catch up per un broker Express riavviato è più veloce rispetto ai broker Standard. La maggiore velocità di applicazione delle patch sui broker Express significa che le interruzioni della pianificazione di tutte le attività del piano di controllo che potresti aver pianificato per il tuo cluster saranno minime.

Come per tutte le applicazioni Apache Kafka, esiste ancora un contratto client-server condiviso per i clienti che si connettono ai broker Express. È ancora fondamentale configurare i clienti per gestire il failover della leadership tra broker. Segui le istruzioni Best practice per i client Apache Kafka per un funzionamento senza intoppi del cluster in ogni momento, anche durante l'applicazione delle patch. Dopo il riavvio del broker, è normale riscontrare errori di disconnessione transitori sui client. Ciò non

influirà sulla produzione e sul consumo, in quanto i broker follower assumeranno la leadership della partizione. I vostri client Apache Kafka eseguiranno automaticamente il failover e inizieranno a inviare richieste ai nuovi broker leader.

Broker offline e failover del client

Kafka consente l'utilizzo di un broker offline; un unico broker offline in un cluster sano ed equilibrato che segue le migliori pratiche non avrà alcun impatto né causerà l'interruzione della produzione o del consumo. Questo perché un altro broker assumerà la guida delle partizioni e perché la libreria dei client di Kafka eseguirà automaticamente il failover e inizierà a inviare richieste ai nuovi broker leader.

Contratto client-server

Ciò si traduce in un contratto condiviso tra la libreria client e il comportamento lato server; il server deve assegnare correttamente uno o più nuovi leader e il client deve cambiare broker per inviare le richieste ai nuovi leader in modo tempestivo.

Kafka utilizza le eccezioni per controllare questo flusso:

Un esempio di procedura

1. Il broker A entra in uno stato offline.
2. Il client Kafka riceve un'eccezione (in genere la disconnessione della rete o not_leader_for_partition).
3. Queste eccezioni fanno sì che il client Kafka aggiorni i propri metadati in modo da conoscere i leader più recenti.
4. Il client Kafka riprende a inviare richieste ai nuovi responsabili delle partizioni su altri broker.

Questo processo richiede in genere meno di 2 secondi con il client Java fornito e le configurazioni predefinite. Gli errori lato client sono dettagliati e ripetitivi, ma non sono motivo di preoccupazione, come indicato dal livello «WARN».

Esempio: eccezione 1

```
10:05:25.306 [kafka-producer-network-thread | producer-1] WARN
o.a.k.c.producer.internals.Sender - [Producer clientId=producer-1] Got
error produce response with correlation id 864845 on topic-partition
msk-test-topic-1-0, retrying (2147483646 attempts left). Error:
NETWORK_EXCEPTION. Error Message: Disconnected from node 2
```

Esempio: eccezione 2

```
10:05:25.306 [kafka-producer-network-thread | producer-1] WARN
o.a.k.c.producer.internals.Sender - [Producer clientId=producer-1] Received
invalid metadata error in produce request on partition msk-test-topic-1-41
due to org.apache.kafka.common.errors.NotLeaderOrFollowerException: For
requests intended only for the leader, this error indicates that the broker
is not the current leader. For requests intended for any replica, this
error indicates that the broker is not a replica of the topic partition..
Going to request metadata update now"
```

I client Kafka risolveranno automaticamente questi errori in genere entro 1 secondo e al massimo 3 secondi. Ciò si presenta come produce/consume una latenza a p99 nelle metriche lato client (in genere millisecondi elevati negli anni 100). Un periodo superiore a questo valore indica in genere un problema relativo alla configurazione del client o al carico del controller sul lato server. Consulta la sezione relativa alla risoluzione dei problemi.

Un failover riuscito può essere verificato controllando l'`BytesInPerSec`aumento delle `LeaderCount` metriche su altri broker, il che dimostra che il traffico e la leadership si sono mossi come previsto. Noterai anche un aumento della `UnderReplicatedPartitions` metrica, previsto quando le repliche sono offline con il broker di shutdown.

Risoluzione dei problemi

Il flusso di cui sopra può essere interrotto interrompendo il contratto client-server. I motivi più comuni del problema includono:

- Configurazione errata o utilizzo errato delle librerie dei client Kafka.
- Comportamenti e bug predefiniti imprevisti con librerie di client di terze parti.
- Controller sovraccarico con conseguente rallentamento dell'assegnazione del leader di partizione.
- Viene eletto un nuovo controller, con conseguente rallentamento dell'assegnazione del leader di partizione.

Per garantire un comportamento corretto in caso di fallimento della leadership, consigliamo di:

- È necessario seguire le migliori pratiche lato server per garantire che il controller broker sia dimensionato in modo appropriato per evitare rallentamenti nell'assegnazione dei dirigenti.
- Le librerie client devono avere i nuovi tentativi abilitati per garantire che il client gestisca il failover.

- Le librerie client devono avere retry.backoff.ms configurato (impostazione predefinita 100) per evitare tempeste. connection/request
- Le librerie client devono impostare request.timeout.ms e delivery.timeout.ms su valori in linea con lo SLA delle applicazioni. Valori più elevati comporteranno un failover più lento per determinati tipi di errore.
- Le librerie client devono garantire che bootstrap.servers contenga almeno 3 broker casuali per evitare un impatto sulla disponibilità sulla scoperta iniziale.
- Alcune librerie client sono di livello inferiore rispetto ad altre e si aspettano che lo sviluppatore dell'applicazione implementi autonomamente la logica dei tentativi e la gestione delle eccezioni. Fate riferimento alla documentazione specifica di Client Lib, ad esempio sull'utilizzo, e assicuratevi che venga seguita la reconnect/retry logica corretta.
- Consigliamo di monitorare la latenza lato client per verificare la produzione, il conteggio delle richieste riuscite e il conteggio degli errori per errori non ripetibili.
- Abbiamo osservato che le vecchie librerie golang e ruby di terze parti rimangono dettagliate durante l'intero periodo di tempo offline del broker, nonostante le richieste di produzione e consumo non ne risentano. Ti consigliamo di monitorare sempre le metriche a livello aziendale, oltre a quelle relative alle richieste di successo e agli errori, per determinare se i log registrano un impatto reale rispetto al rumore.
- I clienti non devono allarmarsi per le eccezioni transitorie per network/not_leader, in quanto sono normali, ininfluenti e previste dal protocollo kafka.
- I clienti non devono attivare gli allarmi in UnderReplicatedPartitions quanto sono normali, ininfluenti e previsti da un singolo broker offline.

Registrazione Amazon MSK

Puoi inviare i log del broker Apache Kafka a uno o più dei seguenti tipi di destinazione: Amazon Logs, Amazon S3 CloudWatch , Amazon Data Firehose. Puoi anche registrare le chiamate API Amazon MSK con AWS CloudTrail.

> **Note**
>
> I log dei broker non sono disponibili sui broker Express.

Log di broker

I log di broker consentono di risolvere i problemi delle applicazioni Apache Kafka e di analizzare le comunicazioni con il cluster MSK. È possibile configurare il cluster MSK nuovo o esistente per fornire i log dei broker a livello Info a uno o più dei seguenti tipi di risorse di destinazione: un gruppo di CloudWatch log, un bucket S3, un flusso di distribuzione Firehose. Tramite Firehose è quindi possibile inviare i dati di registro dal flusso di distribuzione a OpenSearch Service. È necessario creare una risorsa di destinazione prima di configurare il cluster per consegnargli i log del broker. Amazon MSK non crea queste risorse di destinazione se non esistono già. Per informazioni su questi tre tipi di risorse di destinazione e su come crearle, consultare la documentazione seguente:

- CloudWatch Registri Amazon
- Amazon S3
- Amazon Data Firehose

Autorizzazioni richieste

Per configurare una destinazione per i log del broker Amazon MSK, l'identità IAM che utilizzi per le operazioni Amazon MSK deve disporre delle autorizzazioni descritte nella policy AWS politica gestita: Amazon MSKFull Access.

Per eseguire lo streaming dei log di broker a un bucket S3, è richiesta anche l'autorizzazione `s3:PutBucketPolicy`. Per informazioni sulle policy dei bucket S3, consulta la pagina How Do I Add an S3 Bucket Policy? nella Guida per l'utente di Amazon S3. Per informazioni sulle policy IAM in generale, consulta la pagina Access Management nella Guida per l'utente di IAM.

Policy della chiave KMS necessaria per l'utilizzo con i bucket SSE-KMS

Se hai abilitato la crittografia lato server per il tuo bucket S3 utilizzando chiavi AWS KMS gestite (SSE-KMS) con una chiave gestita dal cliente, aggiungi quanto segue alla policy chiave per la tua chiave KMS in modo che Amazon MSK possa scrivere i file del broker nel bucket.

```
{
  "Sid": "Allow Amazon MSK to use the key.",
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "delivery.logs.amazonaws.com"
    ]
```

```
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

Configura i log del broker utilizzando il AWS Management Console

Se stai creando un nuovo cluster, cerca l'intestazione Broker log delivery (Recapito del log del broker) nella sezione Monitoring (Monitoraggio) . Puoi specificare le destinazioni a cui Amazon MSK deve consegnare i log del broker.

Per un cluster esistente, scegli il cluster dall'elenco di cluster, quindi seleziona la scheda Proprietà. Scorri verso il basso fino alla sezione Consegna dei log e scegli il relativo pulsante Modifica. Puoi specificare le destinazioni a cui Amazon MSK deve consegnare i log del broker.

Configurare i log del broker utilizzando il AWS CLI

Quando utilizzi i comandi `create-cluster` o `update-monitoring`, puoi specificare facoltativamente il parametro `logging-info` e passarlo a una struttura JSON come nell'esempio seguente. In questo JSON, tutti e tre i tipi di destinazione sono facoltativi.

```
{
  "BrokerLogs": {
    "S3": {
      "Bucket": "amzn-s3-demo-bucket",
      "Prefix": "ExamplePrefix",
      "Enabled": true
    },
    "Firehose": {
      "DeliveryStream": "ExampleDeliveryStreamName",
      "Enabled": true
    },
    "CloudWatchLogs": {
      "Enabled": true,
      "LogGroup": "ExampleLogGroupName"
    }
```

```
  }
}
```

Configura i log del broker utilizzando l'API

È possibile specificare la `loggingInfo` struttura opzionale nel file JSON che si passa alle operazioni CreateClusteror UpdateMonitoring.

> **Note**
>
> Per impostazione predefinita, quando la registrazione del broker è abilitata, Amazon MSK registra i log di livello `INFO` nelle destinazioni specificate. Tuttavia, gli utenti di Apache Kafka 2.4.X e versioni successive possono impostare dinamicamente il livello di log del broker su uno qualsiasi dei livelli di log log4j. Per informazioni sull'impostazione dinamica del livello di log del broker, consulta la pagina KIP-412: Extend Admin API to support dynamic application log levels. Se imposti dinamicamente il livello di registro su `DEBUG` o`TRACE`, ti consigliamo di utilizzare Amazon S3 o Firehose come destinazione del registro. Se utilizzi CloudWatch Logs come destinazione di log e abiliti `DEBUG` o `TRACE` livelli dinamicamente la registrazione, Amazon MSK può fornire continuamente un campione di log. Ciò può influire in modo significativo sulle prestazioni del broker e deve essere utilizzato solo quando il livello di log `INFO` non è sufficientemente dettagliato da consentire di determinare la causa principale di un problema.

Registra le chiamate API con AWS CloudTrail

> **Note**
>
> AWS CloudTrail i log sono disponibili per Amazon MSK solo quando li usi. Controllo degli accessi IAM

Amazon MSK è integrato con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un utente, un ruolo o un AWS servizio in Amazon MSK. CloudTrail acquisisce le chiamate API come eventi. Le chiamate acquisite includono le chiamate dalla console Amazon MSK e le chiamate di codice alle operazioni API di Amazon MSK. Registra anche le operazioni di Apache Kafka come la creazione e la modifica di argomenti e gruppi.

Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per Amazon MSK. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta effettuata ad Amazon MSK o l'azione Apache Kafka, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, incluso come configurarlo e abilitarlo, consulta la Guida per l'utente.AWS CloudTrail

Informazioni su Amazon MSK in CloudTrail

CloudTrail è abilitato sul tuo account Amazon Web Services al momento della creazione dell'account. Quando si verifica un'attività di evento supportata in un cluster MSK, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli eventi recenti nell'account Amazon Web Services. Per ulteriori informazioni, consulta Visualizzazione di eventi mediante la cronologia eventi di CloudTrail.

Per una registrazione continua degli eventi nell'account Amazon Web Services che includa gli eventi per Amazon MSK, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando si crea un trail nella console, il trail sarà valido in tutte le Regioni . Il trail registra gli eventi di tutte le Regioni nella partizione AWS e distribuisce i file di log nel bucket Amazon S3 specificato. Inoltre, puoi configurare altri servizi Amazon per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei CloudTrail log. Per ulteriori informazioni, consulta gli argomenti seguenti:

- Panoramica della creazione di un trail
- CloudTrail Servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Amazon MSK registra tutte le operazioni di Amazon MSK come eventi nei CloudTrail file di registro. Inoltre, registra le seguenti operazioni di Apache Kafka.

- cluster kafka: DescribeClusterDynamicConfiguration
- ammasso kafka: AlterClusterDynamicConfiguration
- ammasso kafka: CreateTopic

- ammasso kafka: DescribeTopicDynamicConfiguration
- ammasso kafka: AlterTopic
- ammasso kafka: AlterTopicDynamicConfiguration
- ammasso kafka: DeleteTopic

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente root o utente AWS Identity and Access Management (IAM).
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Esempio: voci del file di log di Amazon MSK

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia stack ordinata delle chiamate API pubbliche e delle azioni di Apache Kafka, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra le voci di CloudTrail registro che illustrano le azioni `DescribeCluster` e `DeleteCluster` Amazon MSK.

```
{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "ABCDEF0123456789ABCDE",
        "arn": "arn:aws:iam::012345678901:user/Joe",
        "accountId": "012345678901",
        "accessKeyId": "AIDACKCEVSQ6C2EXAMPLE",
```

```
        "userName": "Joe"
      },
      "eventTime": "2018-12-12T02:29:24Z",
      "eventSource": "kafka.amazonaws.com",
      "eventName": "DescribeCluster",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "aws-cli/1.14.67 Python/3.6.0 Windows/10 botocore/1.9.20",
      "requestParameters": {
        "clusterArn": "arn%3Aaws%3Akafka%3Aus-east-1%3A012345678901%3Acluster
%2Fexamplecluster%2F01234567-abcd-0123-abcd-abcd0123efa-2"
      },
      "responseElements": null,
      "requestID": "bd83f636-fdb5-abcd-0123-157e2fbf2bde",
      "eventID": "60052aba-0123-4511-bcde-3e18dbd42aa4",
      "readOnly": true,
      "eventType": "AwsApiCall",
      "recipientAccountId": "012345678901"
    },
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "ABCDEF0123456789ABCDE",
        "arn": "arn:aws:iam::012345678901:user/Joe",
        "accountId": "012345678901",
        "accessKeyId": "AIDACKCEVSQ6C2EXAMPLE",
        "userName": "Joe"
      },
      "eventTime": "2018-12-12T02:29:40Z",
      "eventSource": "kafka.amazonaws.com",
      "eventName": "DeleteCluster",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "aws-cli/1.14.67 Python/3.6.0 Windows/10 botocore/1.9.20",
      "requestParameters": {
        "clusterArn": "arn%3Aaws%3Akafka%3Aus-east-1%3A012345678901%3Acluster
%2Fexamplecluster%2F01234567-abcd-0123-abcd-abcd0123efa-2"
      },
      "responseElements": {
        "clusterArn": "arn:aws:kafka:us-east-1:012345678901:cluster/
examplecluster/01234567-abcd-0123-abcd-abcd0123efa-2",
        "state": "DELETING"
      },
```

```
      "requestID": "c6bfb3f7-abcd-0123-afa5-293519897703",
      "eventID": "8a7f1fcf-0123-abcd-9bdb-1ebf0663a75c",
      "readOnly": false,
      "eventType": "AwsApiCall",
      "recipientAccountId": "012345678901"
    }
  ]
}
```

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'`kafka-cluster:CreateTopic`azione.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "ABCDEFGH1IJKLMN2P34Q5",
    "arn": "arn:aws:iam::111122223333:user/Admin",
    "accountId": "111122223333",
    "accessKeyId": "CDEFAB1C2UUUUU3AB4TT",
    "userName": "Admin"
  },
  "eventTime": "2021-03-01T12:51:19Z",
  "eventSource": "kafka-cluster.amazonaws.com",
  "eventName": "CreateTopic",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "198.51.100.0/24",
  "userAgent": "aws-msk-iam-auth/unknown-version/aws-internal/3 aws-sdk-java/1.11.970
 Linux/4.14.214-160.339.amzn2.x86_64 OpenJDK_64-Bit_Server_VM/25.272-b10 java/1.8.0_272
 scala/2.12.8 vendor/Red_Hat,_Inc.",
  "requestParameters": {
    "kafkaAPI": "CreateTopics",
    "resourceARN": "arn:aws:kafka:us-east-1:111122223333:topic/IamAuthCluster/3ebafd8e-
dae9-440d-85db-4ef52679674d-1/Topic9"
  },
  "responseElements": null,
  "requestID": "e7c5e49f-6aac-4c9a-a1d1-c2c46599f5e4",
  "eventID": "be1f93fd-4f14-4634-ab02-b5a79cb833d2",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "111122223333"
```

```
}
```

Gestione dei metadati

Amazon MSK supporta Apache ZooKeeper o le modalità di gestione KRaft dei metadati.

Dalla versione 3.7.x di Apache Kafka su Amazon MSK, puoi creare cluster che utilizzano la modalità anziché la modalità. KRaft ZooKeeper KRafti cluster basati su controller all'interno di Kafka per gestire i metadati.

Argomenti

- ZooKeeper modalità
- KRaft modalità

ZooKeeper modalità

Apache ZooKeeper è «un servizio centralizzato per la gestione delle informazioni di configurazione, la denominazione, la sincronizzazione distribuita e la fornitura di servizi di gruppo. Tutti questi tipi di servizi vengono utilizzati in una forma o nell'altra da applicazioni distribuite», incluso Apache Kafka.

Se il tuo cluster utilizza la ZooKeeper modalità, puoi utilizzare i passaggi seguenti per ottenere la stringa di connessione ZooKeeper Apache. Tuttavia, ti consigliamo di utilizzare il `BootstrapServerString` per connetterti al tuo cluster ed eseguire operazioni di amministrazione poiché il `--zookeeper` flag è stato reso obsoleto in Kafka 2.5 ed è stato rimosso da Kafka 3.0.

ZooKeeper Ottenere la stringa di connessione di Apache utilizzando il AWS Management Console

1. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/.
2. La tabella mostra tutti i cluster per la regione corrente in questo account. Scegli il nome di un cluster per visualizzarne la descrizione.
3. Nella pagina Riepilogo del cluster, scegli Visualizza informazioni sul client. Questo mostra i broker bootstrap e la stringa di connessione ZooKeeper Apache.

Ottenere la stringa di connessione Apache usando ZooKeeper il AWS CLI

1. Se l'Amazon Resource Name (ARN) del cluster non è noto, puoi trovarlo elencando tutti i cluster nell'account. Per ulteriori informazioni, consulta the section called “Elenca i cluster”.

2. Per ottenere la stringa di ZooKeeper connessione Apache, insieme ad altre informazioni sul cluster, esegui il comando seguente, sostituendolo *`ClusterArn`* con l'ARN del cluster.

```
aws kafka describe-cluster --cluster-arn ClusterArn
```

L'output di questo comando `describe-cluster` è simile all'esempio JSON seguente.

```
{
    "ClusterInfo": {
        "BrokerNodeGroupInfo": {
            "BrokerAZDistribution": "DEFAULT",
            "ClientSubnets": [
                "subnet-0123456789abcdef0",
                "subnet-2468013579abcdef1",
                "subnet-1357902468abcdef2"
            ],
            "InstanceType": "kafka.m5.large",
            "StorageInfo": {
                "EbsStorageInfo": {
                    "VolumeSize": 1000
                }
            }
        },
        "ClusterArn": "arn:aws:kafka:us-east-1:111122223333:cluster/
testcluster/12345678-abcd-4567-2345-abcdef123456-2",
        "ClusterName": "testcluster",
        "CreationTime": "2018-12-02T17:38:36.75Z",
        "CurrentBrokerSoftwareInfo": {
            "KafkaVersion": "2.2.1"
        },
        "CurrentVersion": "K13V1IB3VIYZZH",
        "EncryptionInfo": {
            "EncryptionAtRest": {
                "DataVolumeKMSKeyId": "arn:aws:kms:us-
east-1:555555555555:key/12345678-abcd-2345-ef01-abcdef123456"
            }
        },
        "EnhancedMonitoring": "DEFAULT",
        "NumberOfBrokerNodes": 3,
        "State": "ACTIVE",
        "ZookeeperConnectString": "10.0.1.101:2018,10.0.2.101:2018,10.0.3.101:2018"
    }
```

```
}
```

L'esempio JSON precedente mostra la chiave `ZookeeperConnectString` nell'output del comando `describe-cluster`. Copia il valore corrispondente a questa chiave e salvalo per utilizzarlo quando è necessario creare un argomento nel cluster.

> **Important**
>
> Il cluster Amazon MSK deve trovarsi nello `ACTIVE` stato in cui è possibile ottenere la stringa di ZooKeeper connessione Apache. Quando un cluster è ancora nello stato `CREATING`, l'output del comando `describe-cluster` non include `ZookeeperConnectString`. In questo caso, occorre attendere alcuni minuti ed eseguire nuovamente `describe-cluster` dopo che il cluster raggiunge lo stato `ACTIVE`.

Ottenere la stringa di ZooKeeper connessione Apache tramite l'API

Per ottenere la stringa di ZooKeeper connessione Apache utilizzando l'API, vedi. DescribeCluster

KRaft modalità

Amazon MSK ha introdotto il supporto per KRaft (Apache Kafka Raft) nella versione 3.7.x di Kafka. La community Apache Kafka si è sviluppata KRaft per sostituire Apache per la gestione dei metadati nei cluster Apache Kafka. ZooKeeper In KRaft modalità, i metadati del cluster vengono propagati all'interno di un gruppo di controller Kafka, che fanno parte del cluster Kafka, anziché tra i nodi. ZooKeeper KRafti controller sono inclusi senza costi aggiuntivi per l'utente e non richiedono alcuna configurazione o gestione aggiuntiva da parte dell'utente. Vedi KIP-500 per ulteriori informazioni su. KRaft

Ecco alcuni punti da tenere in considerazione sulla KRaft modalità su MSK:

- KRaft la modalità è disponibile solo per i nuovi cluster. Non è possibile cambiare modalità di metadati una volta creato il cluster.
- Sulla console MSK, è possibile creare un cluster basato su Kraft scegliendo la versione 3.7.x di Kafka e selezionando la casella di controllo nella finestra di creazione del cluster. KRaft

- Per creare un cluster in KRaft modalità utilizzando l'API `CreateCluster`o le operazioni MSK, è necessario utilizzare come versione. `CreateClusterV2`3.7.x.kraft Utilizza `3.7.x` come versione per creare un cluster in ZooKeeper modalità.
- Il numero di partizioni per broker è lo stesso su KRaft e per i cluster ZooKeeper basati. Tuttavia, KRaft consente di ospitare più partizioni per cluster fornendo più broker in un cluster.
- Non sono necessarie modifiche all'API per utilizzare la KRaft modalità su Amazon MSK. Tuttavia, se i tuoi client utilizzano ancora la stringa di `--zookeeper` connessione oggi, dovresti aggiornarli in modo che utilizzino la stringa di `--bootstrap-server` connessione per connettersi al cluster. Il `--zookeeper` flag è obsoleto nella versione 2.5 di Apache Kafka e viene rimosso a partire dalla versione 3.0 di Kafka. Ti consigliamo quindi di utilizzare le versioni recenti del client Apache Kafka e la stringa di connessione per tutte le connessioni al tuo cluster. `--bootstrap-server`
- ZooKeeper la modalità continua a essere disponibile per tutte le versioni rilasciate in cui zookeeper è supportato anche da Apache Kafka. Vedi Versioni di Apache Kafka supportate i dettagli sulla fine del supporto per le versioni di Apache Kafka e gli aggiornamenti futuri.
- È necessario verificare che tutti gli strumenti utilizzati siano in grado di utilizzare Kafka Admin senza connessioni. APIs ZooKeeper Consulta la procedura aggiornata Usa LinkedIn il Cruise Control per Apache Kafka con Amazon MSK per connettere il cluster a Cruise Control. Cruise Control fornisce anche istruzioni per utilizzare il Cruise Control senza ZooKeeper.
- Non è necessario accedere direttamente ai KRaft controller del cluster per eventuali azioni amministrative. Tuttavia, se si utilizza il monitoraggio aperto per raccogliere le metriche, sono necessari anche gli endpoint DNS dei controller per raccogliere alcune metriche non correlate ai controller sul cluster. È possibile ottenere questi endpoint DNS dalla console MSK o utilizzando l'operazione API. ListNodes Vedi Monitora un cluster MSK Provisioned con Prometheus i passaggi aggiornati per configurare il monitoraggio aperto per i cluster basati. KRaft
- Non sono necessarie CloudWatch metriche aggiuntive per monitorare i cluster in KRaft modalità rispetto ai cluster modali. ZooKeeper MSK gestisce i KRaft controller utilizzati nei cluster.
- È possibile continuare a gestire ACLs utilizzando i cluster in KRaft modalità utilizzando la stringa di connessione. `--bootstrap-server` Non è necessario utilizzare la stringa di `--zookeeper` connessione per gestire ACLs. Consultare Apache Kafka ACLs.
- In KRaft modalità, i metadati del cluster vengono archiviati su KRaft controller all'interno di Kafka e non su nodi esterni. ZooKeeper Pertanto, non è necessario controllare l'accesso ai nodi del controller separatamente come si fa con i nodi. ZooKeeper

Risorse Amazon MSK

Il termine risorse ha due significati in Amazon MSK, a seconda del contesto. Nel contesto di APIs una risorsa c'è una struttura sulla quale è possibile richiamare un'operazione. Per un elenco di queste risorse e delle operazioni che è possibile richiamare su di esse, consulta la sezione Resources nella documentazione di riferimento dell'API di Amazon MSK. Nel contesto di the section called “Controllo degli accessi IAM”, una risorsa è un'entità a cui è possibile consentire o rifiutare l'accesso, come definito nella sezione the section called “Risorse relative alla politica di autorizzazione”.

Versioni di Apache Kafka

Quando si crea un cluster Amazon MSK, specifica quale versione di Apache Kafka desideri utilizzare. Puoi inoltre aggiornare la versione di Apache Kafka di un cluster esistente. Gli argomenti del capitolo ti aiutano a comprendere le tempistiche per il supporto delle versioni di Kafka e i suggerimenti per le migliori pratiche.

Argomenti

- Versioni di Apache Kafka supportate
- Supporto per la versione di Amazon MSK

Versioni di Apache Kafka supportate

Streaming gestito da Amazon per Apache Kafka (Amazon MSK) supporta le seguenti versioni di Apache Kafka e Amazon MSK. La community di Apache Kafka fornisce circa 12 mesi di supporto per una versione successiva alla data di rilascio. Per maggiori dettagli, consulta la politica EOL (end of life) di Apache Kafka.

Versioni di Apache Kafka supportate

Versione Apache Kafka	Data di rilascio di MSK	Data di fine del supporto
1.1.1	--	2024-06-05
2.1.0	--	2024-06-05
2.2.1	31-07-2019	2024-06-08
2.3.1	19-12-2019	2024-06-08

Versione Apache Kafka	Data di rilascio di MSK	Data di fine del supporto
24.1	2020-04-02	2024-06-08
2,41.1	2020-09-09	2024-06-08
2.5.1	2020-09-30	2024-06-08
2,6,0	2020-10-21	2024-09-11
2,6,1	2021-01-19	2024-09-11
2,6,2	2021-04-29	2024-09-11
2,6,3	2021-12-21	2024-09-11
2,7,0	2020-12-29	2024-09-11
2,7,1	2021-05-25	2024-09-11
2,7,2	2021-12-21	2024-09-11
2,80	2021-05-19	2024-09-11
28,1	2022-10-28	2024-09-11
2.8.2 livelli	2022-10-28	2025-01-14
3.1.1	2022-06-22	2024-09-11
32,0	2022-06-22	2024-09-11
3,31	2022-10-26	2024-09-11
3,32	-02	2024-09-11
3,40	2023-05-04	2025-08-04
3,5,1	2023-09-26	2025-10-23
3,6,0	2023-11-16	--

Versione Apache Kafka	Data di rilascio di MSK	Data di fine del supporto
3,7. x	2024-05-29	--
3.8.x	2025-02-20	--
3.9. x	2025-04-21	--
4,0x	2025-05-16	--

Per ulteriori informazioni sulla politica di supporto delle versioni di Amazon MSK, consultaPolitica di supporto delle versioni di Amazon MSK.

Amazon MSK versione 4.0.x

Amazon Managed Streaming for Apache Kafka (Amazon MSK) ora supporta Apache Kafka versione 4.0. Questa versione apporta i più recenti progressi nella gestione e nelle prestazioni dei cluster a MSK Provisioned. Kafka 4.0 introduce un nuovo protocollo di ribilanciamento per i consumatori, ora disponibile a tutti, che aiuta a garantire ribilanciamenti di gruppo più fluidi e rapidi. Inoltre, Kafka 4.0 richiede broker e strumenti per utilizzare Java 17, che offre sicurezza e prestazioni migliorate, include varie correzioni di bug e miglioramenti e rende obsoleta la gestione dei metadati tramite Apache. ZooKeeper

Per maggiori dettagli e un elenco completo di miglioramenti e correzioni di bug, consulta le note di rilascio di Apache Kafka per la versione 4.0.

Amazon MSK versione 3.9.x

Amazon Managed Streaming for Apache Kafka (Amazon MSK) ora supporta Apache Kafka versione 3.9. Questa versione consente di conservare i dati su più livelli quando si disabilita lo storage su più livelli a livello di argomento. Le applicazioni consumer possono continuare a leggere i dati storici dal Remote Log Start Offset (Rx) mantenendo al contempo gli offset di log continui sullo storage locale e remoto.

Per maggiori dettagli e un elenco completo dei miglioramenti e delle correzioni di bug, consulta le note di rilascio di Apache Kafka per la versione 3.9.x.

Amazon MSK versione 3.8.x

Amazon Managed Streaming for Apache Kafka (Amazon MSK) ora supporta Apache Kafka versione 3.8. Ora puoi creare nuovi cluster utilizzando la versione 3.8 con KRAFT o la ZooKeeper modalità per la gestione dei metadati o aggiornare i cluster basati esistenti per utilizzare la versione 3.8. ZooKeeper La versione 3.8 di Apache Kafka include diverse correzioni di bug e nuove funzionalità che migliorano le prestazioni. Le nuove funzionalità chiave includono il supporto per la configurazione del livello di compressione. Ciò consente di ottimizzare ulteriormente le prestazioni quando si utilizzano tipi di compressione come lz4, zstd e gzip, modificando il livello di compressione predefinito.

Per maggiori dettagli e un elenco completo dei miglioramenti e delle correzioni di bug, consulta le note di rilascio di Apache Kafka per la versione 3.8.x.

Apache Kafka versione 3.7.x (con storage su più livelli pronto per la produzione)

La versione 3.7.x di Apache Kafka su MSK include il supporto per Apache Kafka versione 3.7.0. È possibile creare cluster o aggiornare i cluster esistenti per utilizzare la nuova versione 3.7.x. Con questa modifica nella denominazione delle versioni, non è più necessario adottare versioni di patch fix più recenti come la 3.7.1 quando vengono rilasciate dalla community di Apache Kafka. Amazon MSK aggiornerà automaticamente la versione 3.7.x per supportare le future versioni delle patch non appena saranno disponibili. Ciò consente di sfruttare la sicurezza e le correzioni di bug disponibili tramite le versioni patch fix senza attivare un aggiornamento della versione. Queste versioni di patch fix rilasciate da Apache Kafka non compromettono la compatibilità delle versioni e puoi trarre vantaggio dalle nuove versioni di patch fix senza preoccuparti degli errori di lettura o scrittura delle applicazioni client. Assicurati che gli strumenti di automazione dell'infrastruttura, ad esempio CloudFormation, siano aggiornati per tenere conto di questa modifica nella denominazione delle versioni.

Amazon MSK ora supporta la KRaft modalità (Apache Kafka Raft) nella versione 3.7.x di Apache Kafka. Su Amazon MSK, come per i ZooKeeper nodi, KRaft i controller sono inclusi senza costi aggiuntivi e non richiedono alcuna configurazione o gestione aggiuntiva da parte dell'utente. Ora puoi creare cluster in entrambe le KRaft modalità o ZooKeeper modalità su Apache Kafka versione 3.7.x. In modalità Kraft, puoi aggiungere fino a 60 broker per ospitare più partizioni per cluster, senza richiedere un aumento del limite, rispetto alla quota di 30 broker sui cluster basati su ZooKeeper. KRaft modalitàPer ulteriori informazioni su MSK, consulta. KRaft

La versione 3.7.x di Apache Kafka include anche diverse correzioni di bug e nuove funzionalità che migliorano le prestazioni. I miglioramenti principali includono le ottimizzazioni di Leader Discovery per

i client e le opzioni di ottimizzazione del log Segment Flush. Per un elenco completo dei miglioramenti e delle correzioni di bug, consultate le note di rilascio di Apache Kafka per la versione 3.7.0.

Apache Kafka versione 3.6.0 (con archiviazione a più livelli pronta per la produzione)

Per informazioni su Apache Kafka versione 3.6.0 (con archiviazione a più livelli pronta per la produzione), consulta le relative note di rilascio sul sito dei download di Apache Kafka.

Per motivi di stabilità, Amazon MSK continuerà a utilizzare e gestire ZooKeeper per la gestione del quorum in questa versione.

Amazon MSK versione 3.5.1

Amazon Managed Streaming for Apache Kafka (Amazon MSK) ora supporta la versione 3.5.1 di Apache Kafka per cluster nuovi ed esistenti. Apache Kafka 3.5.1 include diverse correzioni di bug e nuove funzionalità che migliorano le prestazioni. Le caratteristiche principali includono l'introduzione di una nuova assegnazione delle partizioni compatibile con i rack per i consumatori. Amazon MSK continuerà a utilizzare e gestire Zookeeper per la gestione del quorum in questa versione. Per un elenco completo dei miglioramenti e delle correzioni di bug, consulta le note di rilascio di Apache Kafka per la versione 3.5.1.

Per informazioni su Apache Kafka versione 3.5.1, consulta le relative note di rilascio sul sito dei download di Apache Kafka.

Amazon MSK versione 3.4.0

Amazon Managed Streaming for Apache Kafka (Amazon MSK) ora supporta Apache Kafka versione 3.4.0 per cluster nuovi ed esistenti. Apache Kafka 3.4.0 include diverse correzioni di bug e nuove funzionalità che migliorano le prestazioni. Le funzionalità principali includono una correzione per migliorare la stabilità da recuperare dalla replica più vicina. Amazon MSK continuerà a utilizzare e gestire Zookeeper per la gestione del quorum in questa versione. Per un elenco completo dei miglioramenti e delle correzioni di bug, consulta le note di rilascio di Apache Kafka per la versione 3.4.0.

Per informazioni su Apache Kafka versione 3.4.0, consulta le relative note di rilascio sul sito dei download di Apache Kafka.

Amazon MSK versione 3.3.2

Amazon Managed Streaming for Apache Kafka (Amazon MSK) ora supporta la versione 3.3.2 di Apache Kafka per cluster nuovi ed esistenti. Apache Kafka 3.3.2 include diverse correzioni di bug e

nuove funzionalità che migliorano le prestazioni. Le funzionalità principali includono una correzione per migliorare la stabilità da recuperare dalla replica più vicina. Amazon MSK continuerà a utilizzare e gestire Zookeeper per la gestione del quorum in questa versione. Per un elenco completo dei miglioramenti e delle correzioni di bug, consulta le note di rilascio di Apache Kafka per la versione 3.3.2.

Per informazioni su Apache Kafka versione 3.3.2, consulta le relative note di rilascio sul sito dei download di Apache Kafka.

Amazon MSK versione 3.3.1

Amazon Managed Streaming for Apache Kafka (Amazon MSK) ora supporta la versione 3.3.1 di Apache Kafka per cluster nuovi ed esistenti. Apache Kafka 3.3.1 include diverse correzioni di bug e nuove funzionalità che migliorano le prestazioni. Alcune delle funzionalità principali includono miglioramenti alle metriche e al partizionatore. Per motivi di stabilità, Amazon MSK continuerà a utilizzare e gestire ZooKeeper per la gestione del quorum in questa versione. Per un elenco completo dei miglioramenti e delle correzioni di bug, consultate le note di rilascio di Apache Kafka per la versione 3.3.1.

Per informazioni su Apache Kafka versione 3.3.1, consulta le relative note di rilascio sul sito dei download di Apache Kafka.

Amazon MSK versione 3.1.1

Amazon Managed Streaming for Apache Kafka (Amazon MSK) ora supporta le versioni 3.1.1 e 3.2.0 di Apache Kafka per cluster nuovi ed esistenti. Apache Kafka 3.1.1 e Apache Kafka 3.2.0 includono diverse correzioni di bug e nuove funzionalità che migliorano le prestazioni. Alcune delle funzionalità principali includono miglioramenti alle metriche e l'uso dell'argomento. IDs MSK continuerà a utilizzare e gestire Zookeeper per la gestione del quorum in questa versione per motivi di stabilità. Per un elenco completo dei miglioramenti e delle correzioni di bug, consultate le note di rilascio di Apache Kafka per 3.1.1 e 3.2.0.

Per informazioni sulle versioni 3.1.1 e 3.2.0 di Apache Kafka, consultate le relative note di rilascio 3.2.0 e le note di rilascio 3.1.1 sul sito di download di Apache Kafka.

Archiviazione a più livelli Amazon MSK versione 2.8.2.tiered

Questa versione è una versione solo per Amazon MSK di Apache Kafka versione 2.8.2 ed è compatibile con i client open source Apache Kafka.

La versione 2.8.2.tiered contiene funzionalità di storage su più livelli compatibili con quelle introdotte in KIP-405 per Apache Kafka. APIs Per ulteriori informazioni sulla funzionalità di archiviazione a più livelli di Amazon MSK, consulta la sezione Storage su più livelli per broker Standard.

Apache Kafka versione 2.5.1

La versione 2.5.1 di Apache Kafka include diverse correzioni di bug e nuove funzionalità, tra cui la crittografia in transito per Apache e i client di amministrazione. ZooKeeper Amazon MSK fornisce ZooKeeper endpoint TLS, che possono essere interrogati durante l'operazione. DescribeCluster

L'output dell' DescribeClusteroperazione include il `ZookeeperConnectStringTls` nodo, che elenca gli endpoint TLS zookeeper.

L'esempio seguente mostra il nodo `ZookeeperConnectStringTls` della risposta per l'operazione `DescribeCluster`:

```
"ZookeeperConnectStringTls": "z-3.awskafkatutorialc.abcd123.c3.kafka.us-
east-1.amazonaws.com:2182,z-2.awskafkatutorialc.abcd123.c3.kafka.us-
east-1.amazonaws.com:2182,z-1.awskafkatutorialc.abcd123.c3.kafka.us-
east-1.amazonaws.com:2182"
```

Per informazioni sull'utilizzo della crittografia TLS con ZooKeeper, consulta la sezione Utilizzo della sicurezza TLS con Apache ZooKeeper.

Per ulteriori informazioni su Apache Kafka versione 2.5.1, consulta le relative note di rilascio sul sito dei download di Apache Kafka.

Versione di correzione dei bug Amazon MSK 2.4.1.1

Questa versione è una versione di correzione dei bug di Apache Kafka 2.4.1 disponibile solo per Amazon MSK. Questa versione di correzione contiene una correzione per KAFKA-9752, un problema raro che causa il continuo ribilanciamento dei gruppi di consumatori e la permanenza nello stato `PreparingRebalance`. Questo problema riguarda i cluster che eseguono Apache Kafka versioni 2.3.1 e 2.4.1. Questa versione contiene una correzione prodotta dalla comunità disponibile nella versione 2.5.0 di Apache Kafka.

> **Note**
>
> I cluster Amazon MSK che eseguono la versione 2.4.1.1 sono compatibili con qualsiasi client Apache Kafka compatibile con la versione 2.4.1 di Apache Kafka.

Se preferisci usare Apache Kafka 2.4.1, ti consigliamo di utilizzare la versione 2.4.1.1 di correzione dei bug MSK per i nuovi cluster Amazon MSK. Per incorporare questa correzione, puoi aggiornare i cluster esistenti che eseguono Apache Kafka versione 2.4.1 a questa versione. Per informazioni sull'aggiornamento di un cluster esistente, consulta la sezione Aggiorna la versione di Apache Kafka.

Per risolvere questo problema senza aggiornare il cluster alla versione 2.4.1.1, consulta la sezione Gruppo di consumatori bloccato nello stato `PreparingRebalance` della guida Risolvi i problemi del tuo cluster Amazon MSK.

Apache Kafka versione 2.4.1 (usa invece 2.4.1.1)

> **Note**
>
> Non è più possibile creare un cluster MSK con la versione 2.4.1 di Apache Kafka. In alternativa, è possibile utilizzare Versione di correzione dei bug Amazon MSK 2.4.1.1 con client compatibili con la versione 2.4.1 di Apache Kafka. Se disponi già di un cluster MSK con Apache Kafka versione 2.4.1, ti consigliamo di aggiornarlo per utilizzare invece la versione 2.4.1.1 di Apache Kafka.

KIP-392 è una delle principali proposte di miglioramento di Kafka incluse nella versione 2.4.1 di Apache Kafka. Questo miglioramento consente ai consumatori di recuperare dati dalla replica più vicina. Per utilizzare questa caratteristica, imposta `client.rack` nelle proprietà consumatore sull'ID della zona di disponibilità del consumatore. Un esempio di ID di zona di disponibilità è `use1-az1`. Amazon MSK imposta `broker.rack` le zone IDs di disponibilità dei broker. Inoltre, devi impostare la proprietà di configurazione `replica.selector.class` su `org.apache.kafka.common.replica.RackAwareReplicaSelector`, che è un'implementazione di consapevolezza rack fornita da Apache Kafka.

Quando utilizzi questa versione di Apache Kafka, i parametri nel livello di monitoraggio `PER_TOPIC_PER_BROKER` vengono visualizzati solo dopo che i valori diventano diversi da zero per la prima volta. Per ulteriori informazioni, consulta the section called “Monitoraggio del livello `PER_TOPIC_PER_BROKER`”.

Per informazioni su come trovare la zona di disponibilità IDs, consulta AZ IDs for Your Resource nella guida per l' AWS Resource Access Manager utente.

Per informazioni sull'impostazione delle proprietà di configurazione, consulta the section called “Configurazione del broker”.

Per ulteriori informazioni su KIP-392, consulta Allow Consumers to Fetch from Closest Replica nelle pagine di Confluence.

Per ulteriori informazioni su Apache Kafka versione 2.4.1, consulta le relative note di rilascio sul sito dei download di Apache Kafka.

Supporto per la versione di Amazon MSK

Questo argomento descrive Politica di supporto delle versioni di Amazon MSK e la procedura perAggiorna la versione di Apache Kafka. Se stai aggiornando la tua versione di Kafka, segui le migliori pratiche descritte in. Procedure consigliate per gli aggiornamenti delle versioni

Argomenti

- Politica di supporto delle versioni di Amazon MSK
- Aggiorna la versione di Apache Kafka
- Procedure consigliate per gli aggiornamenti delle versioni

Politica di supporto delle versioni di Amazon MSK

Questa sezione descrive la politica di supporto per le versioni di Kafka supportate da Amazon MSK.

- Tutte le versioni di Kafka sono supportate fino al raggiungimento della data di fine del supporto. Per informazioni dettagliate sulle date di fine del supporto, consulta. Versioni di Apache Kafka supportate Aggiorna il tuo cluster MSK alla versione di Kafka consigliata o alla versione successiva prima della data di fine del supporto. Per dettagli sull'aggiornamento della versione di Apache Kafka, consulta. Aggiorna la versione di Apache Kafka Un cluster che utilizza una versione di Kafka dopo la data di fine del supporto viene aggiornato automaticamente alla versione Kafka consigliata. Gli aggiornamenti automatici possono avvenire in qualsiasi momento dopo la data di fine del supporto. Non riceverai alcuna notifica prima dell'aggiornamento.
- MSK eliminerà gradualmente il supporto per i cluster di nuova creazione che utilizzano versioni di Kafka con date di fine supporto pubblicate.

Aggiorna la versione di Apache Kafka

È possibile aggiornare un cluster MSK esistente a una versione più recente di Apache Kafka.

Note

- Non è possibile aggiornare un cluster MSK esistente da una versione ZooKeeper basata su Apache Kafka a una versione più recente che utilizza o richiede la modalità. KRaft Invece, per aggiornare il tuo cluster, crea un nuovo cluster MSK con una versione di Kafka KRaft supportata e migra i dati e i carichi di lavoro dal vecchio cluster.
- Amazon MSK aggiorna solo il software del server. Non aggiorna i tuoi clienti.
- Non è possibile effettuare il downgrade di un cluster MSK esistente a una versione precedente di Apache Kafka.

Quando aggiorni la versione Apache Kafka di un cluster MSK, controlla anche il software sul lato client per assicurarti che la versione consenta di utilizzare le funzionalità della nuova versione di Apache Kafka del cluster.

Per informazioni su come rendere un cluster altamente disponibile durante un aggiornamento, consulta. the section called “Creazione di cluster a disponibilità elevata”

Aggiornare la versione di Apache Kafka utilizzando il AWS Management Console

1. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/.
2. Nella barra di navigazione, scegli la regione in cui hai creato il cluster MSK.
3. Scegli il cluster MSK che desideri aggiornare.
4. Nella scheda Proprietà, scegli Aggiorna nella sezione relativa alla versione di Apache Kafka.
5. Nella sezione relativa alla versione di Apache Kafka, procedi come segue:

    a. Nell'elenco a discesa Scegli la versione di Apache Kafka, scegli la versione a cui desideri eseguire l'aggiornamento. Ad esempio, scegli **`3.9.x`**.

    b. (Facoltativo) Scegli Compatibilità tra le versioni per visualizzare la compatibilità tra la versione corrente del cluster e la versione a cui desideri eseguire l'aggiornamento. Quindi, scegli Scegli per procedere o scegli Annulla.

    c. Scegli la casella di controllo Aggiorna la configurazione del cluster per applicare automaticamente una nuova revisione della configurazione di Kafka compatibile con la versione aggiornata. Ciò garantisce la compatibilità e abilita nuove funzionalità o miglioramenti della versione aggiornata. Tuttavia, saltalo se desideri mantenere le configurazioni personalizzate esistenti.

d. Seleziona Upgrade (Aggiorna).

Aggiorna la versione di Apache Kafka usando AWS CLI

1. Esegui il comando seguente, sostituendolo *ClusterArn* con l'Amazon Resource Name (ARN) che hai ottenuto quando hai creato il cluster. Se non disponi dell'ARN per il cluster, puoi trovarlo elencando tutti i cluster. Per ulteriori informazioni, consulta the section called “Elenca i cluster”.

```
aws kafka get-compatible-kafka-versions --cluster-arn ClusterArn
```

L'output di questo comando include un elenco delle versioni di Apache Kafka a cui è possibile aggiornare il cluster. Il risultato sembra l'esempio seguente.

```
{
    "CompatibleKafkaVersions": [
        {
            "SourceVersion": "2.2.1",
            "TargetVersions": [
                "2.3.1",
                "2.4.1",
                "2.4.1.1",
                "2.5.1"
            ]
        }
    ]
}
```

2. Esegui il comando seguente, sostituendolo *ClusterArn* con l'Amazon Resource Name (ARN) che hai ottenuto quando hai creato il cluster. Se non disponi dell'ARN per il cluster, puoi trovarlo elencando tutti i cluster. Per ulteriori informazioni, consulta the section called “Elenca i cluster”.

   Sostituisci *Current-Cluster-Version* con la versione corrente del cluster. Perché *TargetVersion* è possibile specificare una qualsiasi delle versioni di destinazione dall'output del comando precedente.

> **Important**
>
> Le versioni del cluster non sono interi semplici. Per trovare la versione corrente del cluster, usa l'DescribeClusteroperazione o il comando AWS CLI describe-cluster. Una versione di esempio è `KTVPDKIKX0DER`.

```
aws kafka update-cluster-kafka-version --cluster-arn ClusterArn --current-
version Current-Cluster-Version --target-kafka-version TargetVersion
```

L'output del comando precedente è simile al JSON seguente.

```
{

    "ClusterArn": "arn:aws:kafka:us-east-1:012345678012:cluster/exampleClusterName/
abcdefab-1234-abcd-5678-cdef0123ab01-2",
    "ClusterOperationArn": "arn:aws:kafka:us-east-1:012345678012:cluster-
operation/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2/0123abcd-
abcd-4f7f-1234-9876543210ef"
}
```

3. Per ottenere il risultato dell'`update-cluster-kafka-version`operazione, esegui il comando seguente, sostituendolo *`ClusterOperationArn`* con l'ARN ottenuto nell'output del `update-cluster-kafka-version` comando.

```
aws kafka describe-cluster-operation --cluster-operation-arn ClusterOperationArn
```

L'output di questo comando `describe-cluster-operation` è simile all'esempio JSON seguente.

```
{
    "ClusterOperationInfo": {
        "ClientRequestId": "62cd41d2-1206-4ebf-85a8-dbb2ba0fe259",
        "ClusterArn": "arn:aws:kafka:us-east-1:012345678012:cluster/
exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2",
        "CreationTime": "2021-03-11T20:34:59.648000+00:00",
        "OperationArn": "arn:aws:kafka:us-east-1:012345678012:cluster-
operation/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2/0123abcd-
abcd-4f7f-1234-9876543210ef",
```

```
        "OperationState": "UPDATE_IN_PROGRESS",
        "OperationSteps": [
            {
                "StepInfo": {
                    "StepStatus": "IN_PROGRESS"
                },
                "StepName": "INITIALIZE_UPDATE"
            },
            {
                "StepInfo": {
                    "StepStatus": "PENDING"
                },
                "StepName": "UPDATE_APACHE_KAFKA_BINARIES"
            },
            {
                "StepInfo": {
                    "StepStatus": "PENDING"
                },
                "StepName": "FINALIZE_UPDATE"
            }
        ],
        "OperationType": "UPDATE_CLUSTER_KAFKA_VERSION",
        "SourceClusterInfo": {
            "KafkaVersion": "2.4.1"
        },
        "TargetClusterInfo": {
            "KafkaVersion": "2.6.1"
        }
    }
}
```

Se il valore di `OperationState` è `UPDATE_IN_PROGRESS`, attendi qualche minuto, quindi esegui nuovamente il comando `describe-cluster-operation`. Al termine dell'operazione, il valore di `OperationState` diventa `UPDATE_COMPLETE`. Poiché il tempo necessario ad Amazon MSK per completare l'operazione varia, potrebbe essere necessario eseguire ripetutamente il controllo fino al completamento dell'operazione.

Aggiorna la versione di Apache Kafka utilizzando l'API

1. Richiama l'GetCompatibleKafkaVersionsoperazione per ottenere un elenco delle versioni di Apache Kafka a cui è possibile aggiornare il cluster.

2. Richiama l'UpdateClusterKafkaVersionoperazione per aggiornare il cluster a una delle versioni compatibili di Apache Kafka.

Procedure consigliate per gli aggiornamenti delle versioni

Per garantire la continuità del client durante l'aggiornamento progressivo eseguito come parte del processo di aggiornamento della versione di Kafka, rivedi la configurazione dei tuoi client e gli argomenti di Apache Kafka come segue:

- Imposta il fattore di replica dell'argomento (RF) su un valore minimo di per i cluster Two-AZ e un valore minimo di 2 per i cluster Three-AZ. 3 Un valore RF di 2 può portare a partizioni offline durante l'applicazione delle patch.
- Imposta il numero minimo di repliche in-sync (miniSR) su un valore massimo di 1 in meno rispetto al tuo Replication Factor (RF), che è. `miniISR = (RF) - 1` Ciò garantisce che il set di repliche delle partizioni possa tollerare che una replica sia offline o poco replicata.
- Configura i client per utilizzare più stringhe di connessione del broker. La presenza di più broker nella stringa di connessione di un client consente il failover se uno specifico broker che supporta il client I/O inizia a ricevere le patch. Per informazioni su come ottenere una stringa di connessione con più broker, consulta Ottenere i broker bootstrap per un cluster Amazon MSK.
- Ti consigliamo di aggiornare i client che si connettono alla versione consigliata o superiore per beneficiare delle funzionalità disponibili nella nuova versione. Gli upgrade dei client non sono soggetti alle date di fine del ciclo di vita (EOL) della versione Kafka del cluster MSK e non è necessario che vengano completati entro la data di fine del ciclo di vita. Apache Kafka fornisce una politica di compatibilità dei client bidirezionale che consente ai client più vecchi di lavorare con cluster più recenti e viceversa.
- È probabile che i client Kafka che utilizzano le versioni 3.x.x abbiano le seguenti impostazioni predefinite: e. `acks=all` `enable.idempotence=true` `acks=all`è diverso dall'impostazione predefinita precedente di `acks=1` e offre una maggiore durabilità assicurando che tutte le repliche sincronizzate riconoscano la richiesta di produzione. Analogamente, l'impostazione predefinita per `enable.idempotence` era precedente. `false` La modifica all'`enable.idempotence=true`impostazione predefinita riduce la probabilità di messaggi duplicati. Queste modifiche sono considerate impostazioni di best practice e possono introdurre una piccola quantità di latenza aggiuntiva che rientra nei normali parametri di prestazione.
- Usa la versione consigliata di Kafka per creare nuovi cluster MSK. L'utilizzo della versione consigliata di Kafka consente di sfruttare le funzionalità più recenti di Kafka e MSK.

Risolvi i problemi del tuo cluster Amazon MSK

Le seguenti informazioni consentono di semplificare la risoluzione dei problemi che si potrebbero verificare con il cluster Amazon MSK. Puoi anche pubblicare il problema in AWS re:Post. Per la risoluzione dei problemi di Amazon MSK Replicator, consulta. Risolvete i problemi relativi a MSK Replicator

Argomenti

La sostituzione del volume causa la saturazione del disco a causa del sovraccarico della replica

In caso di guasto hardware non pianificato del volume, Amazon MSK può sostituire il volume con una nuova istanza. Kafka ripopola il nuovo volume replicando le partizioni di altri broker del cluster. Una volta che le partizioni sono state replicate e recuperate, sono idonee per l'iscrizione alla leadership e all'In-Sync Replica (ISR).

Problema

In un broker che si sta riprendendo dalla sostituzione dei volumi, alcune partizioni di dimensioni diverse potrebbero tornare online prima di altre. Ciò può essere problematico in quanto tali partizioni possono servire il traffico proveniente dallo stesso broker che sta ancora recuperando (replicando) altre partizioni. Questo traffico di replica a volte può saturare i limiti di throughput del volume sottostanti, che nel caso predefinito sono 250 MiB al secondo. Quando si verifica questa saturazione, tutte le partizioni già interessate ne risentono, con conseguente latenza all'interno del cluster per tutti i broker che condividono ISR con quelle partizioni interessate (non solo le partizioni leader dovute agli ack remoti). `acks=all` Questo problema è più comune nei cluster più grandi che hanno un numero maggiore di partizioni di dimensioni variabili.

Raccomandazione

- Per migliorare la I/O postura di replica, assicuratevi che siano state adottate le migliori impostazioni dei thread.
- Per ridurre la probabilità di saturazione del volume sottostante, abilita lo storage fornito con un throughput più elevato. Un valore minimo di throughput pari a 500 MiB/s è consigliato per i casi di replica con throughput elevato, ma il valore effettivo necessario varia a seconda del throughput e del caso d'uso. Esegui il provisioning del throughput di storage per i broker Standard in un cluster Amazon MSK.
- Per ridurre al minimo la pressione di replica, `num.replica.fetchers` abbassare al valore predefinito di2.

Gruppo di consumatori bloccato nello stato **`PreparingRebalance`**

Se uno o più gruppi di consumatori sono bloccati in uno stato di ribilanciamento perpetuo, la causa potrebbe essere il problema KAFKA-9752 di Apache Kafka, che riguarda le versioni 2.3.1 e 2.4.1 di Apache Kafka.

Per risolvere questo problema, ti consigliamo di aggiornare il cluster alla versione Versione di correzione dei bug Amazon MSK 2.4.1.1, che contiene una correzione per questo problema. Per informazioni sull'aggiornamento di un cluster esistente alla versione 2.4.1.1 di correzione dei bug di Amazon MSK, consulta la pagina Aggiorna la versione di Apache Kafka.

Le soluzioni alternative per risolvere questo problema senza aggiornare il cluster alla versione di correzione del bug Amazon MSK 2.4.1.1 consistono nell'impostare i client Kafka in modo da utilizzare Protocollo di iscrizione statico oppure Identificazione e riavvio il nodo dei broker di coordinamento del gruppo di consumatori bloccato.

Implementazione del protocollo di iscrizione statico

Per implementare il protocollo di iscrizione statico nei client, procedi come indicato di seguito:

1. Imposta la proprietà `group.instance.id` della configurazione dei consumatori Kafka su una stringa statica che identifica il consumatore nel gruppo.
2. Assicurati che le altre istanze della configurazione siano aggiornate in modo da utilizzare la stringa statica.
3. Implementa le modifiche ai tuoi consumatori Kafka.

L'utilizzo del protocollo di iscrizione statico è più efficace se il timeout della sessione nella configurazione client è impostato su una durata che consenta al consumatore di ripristinare il sistema senza innescare prematuramente un ribilanciamento del gruppo di consumatori. Ad esempio, se l'applicazione consumatore può tollerare 5 minuti di indisponibilità, un valore ragionevole per il timeout della sessione sarebbe 4 minuti anziché il valore predefinito di 10 secondi.

> **Note**
>
> L'utilizzo del protocollo di iscrizione statico riduce solamente la probabilità di riscontrare questo problema. È possibile che questo problema si verifichi ancora anche quando si utilizza il protocollo di iscrizione statico.

Riavvio del nodo dei broker di coordinamento

Per riavviare il nodo dei broker di coordinamento, procedi come segue:

1. Identifica il coordinatore del gruppo utilizzando il comando `kafka-consumer-groups.sh`.
2. Riavvia il coordinatore del gruppo di consumatori bloccato utilizzando l'azione RebootBrokerAPI.

Errore nell'invio dei log del broker ad Amazon CloudWatch Logs

Quando provi a configurare il tuo cluster per inviare i log del broker ad Amazon CloudWatch Logs, potresti ottenere una delle due eccezioni.

Se viene restituita un'eccezione `InvalidInput.LengthOfCloudWatchResourcePolicyLimitExceeded`, riprova utilizzando i gruppi di log che iniziano con `/aws/vendedlogs/`. Per ulteriori informazioni, consulta la pagina Enabling Logging from Certain Amazon Web Services.

Se ricevi un'`InvalidInput.NumberOfCloudWatchResourcePoliciesLimitExceeded`eccezione, scegli una policy Amazon CloudWatch Logs esistente nel tuo account e aggiungi il seguente codice JSON.

```
{"Sid":"AWSLogDeliveryWrite","Effect":"Allow","Principal":
{"Service":"delivery.logs.amazonaws.com"},"Action":
["logs:CreateLogStream","logs:PutLogEvents"],"Resource":["*"]}
```

Se provi ad aggiungere il codice JSON sopra riportato a una policy esistente ma ricevi un errore che indica che hai raggiunto la lunghezza massima per la policy che hai scelto, prova ad aggiungere il codice JSON a un'altra delle tue politiche Amazon Logs. CloudWatch Dopo aver aggiunto il codice JSON a una policy esistente, prova ancora una volta a configurare la distribuzione dei log del broker ad Amazon Logs. CloudWatch

Nessun gruppo di sicurezza predefinito

Se cerchi di creare un cluster e ricevi un errore che indica che non esiste un gruppo di sicurezza predefinito, è possibile che il VPC che stai utilizzando sia stato condiviso con te. Chiedi all'amministratore di concedere l'autorizzazione per descrivere i gruppi di sicurezza in questo VPC e riprova. Per un esempio di policy che consente questa azione, consulta Amazon EC2: consente la gestione dei gruppi di EC2 sicurezza associati a un VPC specifico, a livello di programmazione e nella console.

I cluster sono bloccati nello stato CREATING

A volte la creazione del cluster può richiedere fino a 30 minuti. Attendi 30 minuti e controlla nuovamente lo stato del cluster.

Lo stato del cluster passa da CREATING a FAILED

Prova a creare nuovamente il cluster.

Lo stato del cluster è ACTIVE ma i produttori non possono inviare dati o i consumatori non possono ricevere dati

- Se la creazione del cluster va a buon fine (lo stato del cluster è `ACTIVE`), ma non è possibile inviare o ricevere dati, assicurati che le applicazioni produttore e consumatore dispongano dell'accesso al cluster. Per ulteriori informazioni, consulta le linee guida in the section called “Crea una macchina client”.

- Se i produttori e i consumatori dispongono dell'accesso al cluster ma si verificano ancora problemi nella produzione e nel consumo di dati, è possibile che la causa sia riconducibile a KAFKA-7697, che influenza Apache Kafka versione 2.1.0 e può condurre a un deadlock in uno o più broker. Valuta la possibilità di eseguire la migrazione ad Apache Kafka 2.2.1, che non è influenzato da questo bug. Per informazioni sulla migrazione, consulta the section called “Migrazione al cluster Amazon MSK”.

AWS CLI non riconosce Amazon MSK

Se lo hai AWS CLI installato, ma non riconosce i comandi di Amazon MSK, esegui l'upgrade AWS CLI alla versione più recente. Per istruzioni dettagliate su come aggiornare AWS CLI, consulta Installazione di AWS Command Line Interface. Per informazioni su come utilizzare per AWS CLI eseguire i comandi Amazon MSK, consultathe section called “Caratteristiche e concetti chiave”.

Le partizioni vengono messe offline o le repliche non sono sincronizzate

Questi sintomi possono essere causati da spazio su disco insufficiente. Consultare the section called “Lo spazio su disco è insufficiente”.

Lo spazio su disco è insufficiente

Vedere le best practice seguenti per gestire lo spazio su disco: the section called “Monitoraggio dello spazio su disco” e the section called “Regolazione dei parametri di conservazione dei dati”.

La memoria è insufficiente

Se il parametro `MemoryUsed` diventa alto o il parametro `MemoryFree` diventa basso, ciò non significa che ci sia un problema. Apache Kafka è progettato per utilizzare e gestire in maniera ottimale la massima quantità di memoria.

Il produttore ottiene NotLeaderForPartitionException

Questo è spesso un errore temporaneo. Impostare il parametro di configurazione `retries` del produttore su un valore più alto del valore corrente.

Partizioni sottoreplicate (URP) superiori a zero

Il parametro `UnderReplicatedPartitions` è importante da monitorare. In un cluster MSK integro, il valore di questo parametro è 0. Se è maggiore di zero, il motivo potrebbe essere uno dei seguenti.

- Se `UnderReplicatedPartitions` presenta un picco, è possibile che non sia stato effettuato il provisioning del cluster alle dimensioni corrette per gestire il traffico in entrata e in uscita. Consultare the section called “Best practice per broker standard”.
- Se `UnderReplicatedPartitions` è costantemente maggiore di 0, anche durante i periodi di traffico limitato, il problema potrebbe essere dovuto al fatto che hai impostato delle restrizioni ACLs che non garantiscono l'accesso all'argomento ai broker. Per replicare le partizioni, i broker devono disporre dell'autorizzazione per gli argomenti READ e DESCRIBE. L'argomento DESCRIBE viene concesso per impostazione predefinita con l'autorizzazione READ. Per informazioni sull'impostazione ACLs, consulta Autorizzazione e ACLs nella documentazione di Apache Kafka.

Il cluster ha argomenti chiamati __amazon_msk_canary e __amazon_msk_canary_state

Potresti notare che il tuo cluster MSK ha un argomento con il nome `__amazon_msk_canary` e un altro con il nome `__amazon_msk_canary_state`. Si tratta di argomenti interni che Amazon MSK crea e utilizza per i parametri diagnostici e di salute dei cluster. Questi argomenti sono di dimensioni trascurabili e non possono essere eliminati.

La replica delle partizioni ha esito negativo

Assicurati di non aver impostato ACLs CLUSTER_ACTIONS.

Impossibile accedere al cluster con accesso pubblico attivato

Se il cluster ha attivato l'accesso pubblico, ma non riesci ancora ad accedervi da Internet, esegui i passaggi seguenti:

1. Assicurati che le regole in entrata del gruppo di sicurezza del cluster consentano il tuo indirizzo IP e la porta del cluster. Per un elenco dei numeri di porta del cluster, consulta la pagina the

section called “Informazioni sulle porte”. Assicurati inoltre che le regole in uscita del gruppo di sicurezza consentano le comunicazioni in uscita. Per ulteriori informazioni sui gruppi di sicurezza e le rispettive regole in entrata e in uscita, consulta la pagina Security groups for your VPC nella Guida per l'utente di Amazon VPC.

2. Assicurati che il tuo indirizzo IP e la porta del cluster siano consentiti nelle regole in entrata dell'ACL della rete VPC del cluster. A differenza dei gruppi di sicurezza, le reti sono prive di ACLs stato. Ciò significa che è necessario configurarne le regole in entrata e in uscita. Nelle regole in uscita, consenti tutto il traffico (intervallo di porte: 0-65535) verso il tuo indirizzo IP. Per ulteriori informazioni, consulta la pagina Add and delete rules nella Guida per l'utente di Amazon VPC.
3. Assicurati di utilizzare la stringa bootstrap-brokers ad accesso pubblico per accedere al cluster. Un cluster MSK con accesso pubblico attivato ha due diverse stringhe bootstrap-brokers, una per l'accesso pubblico e una per l'accesso dall'interno di AWS. Per ulteriori informazioni, consulta the section called “Ottieni i broker bootstrap usando il AWS Management Console”.

Impossibile accedere al cluster dall'interno AWS: problemi di rete

Se disponi di un'applicazione Apache Kafka che non è in grado di comunicare correttamente con un cluster MSK, inizia eseguendo il seguente test di connettività.

1. Utilizzare uno dei metodi descritti in the section called “Ottieni i broker bootstrap” per ottenere gli indirizzi dei broker bootstrap.
2. Nel comando seguente *`bootstrap-broker`* sostituiscilo con uno degli indirizzi del broker che hai ottenuto nel passaggio precedente. Sostituire *`port-number`* con 9094 se il cluster è configurato per utilizzare l'autenticazione TLS. Se il cluster non utilizza l'autenticazione TLS, *`port-number`* sostituiscila con 9092. Eseguire il comando dal computer client.

   ```
   telnet bootstrap-broker port-number
   ```

   Dove il numero di porta è:

   - 9094 se il cluster è configurato per utilizzare l'autenticazione TLS.
   - 9092 Se il cluster non utilizza l'autenticazione TLS.
   - È necessario un numero di porta diverso se l'accesso pubblico è abilitato.

   Eseguire il comando dal computer client.

3. Ripetere il comando precedente per tutti i broker bootstrap.

Se la macchina client è in grado di accedere ai broker, significa che non ci sono problemi di connettività. In questo caso, eseguire il comando seguente per verificare se il client Apache Kafka è configurato correttamente. Per ottenerlo*bootstrap-brokers*, usa uno dei metodi descritti inthe section called “Ottieni i broker bootstrap”. Sostituiscilo *topic* con il nome del tuo argomento.

```
<path-to-your-kafka-installation>/bin/kafka-console-producer.sh --broker-
list bootstrap-brokers --producer.config client.properties --topic topic
```

Se il comando precedente va a buon fine, significa che il client è configurato correttamente. Se non è ancora possibile produrre e consumare da un'applicazione, eseguire il debug del problema a livello di applicazione.

Se il computer client non è in grado di accedere ai broker, consulta le seguenti sottosezioni per una guida basata sulla configurazione del computer client.

EC2 Client Amazon e cluster MSK nello stesso VPC

Se il computer client si trova nello stesso VPC del cluster MSK, assicurati che il gruppo di sicurezza del cluster disponga di una regola in entrata che accetta il traffico dal gruppo di sicurezza del computer client. Per informazioni sull'impostazione di queste regole, consulta Regole del gruppo di sicurezza. Per un esempio di come accedere a un cluster da un' EC2 istanza Amazon che si trova nello stesso VPC del cluster, vedi. the section called “Inizia a usare”

EC2 Client Amazon e cluster MSK in diversi VPCs

Se il computer client e il cluster si trovano in due ambienti diversi VPCs, verifica quanto segue:

- I due VPCs sono peer-to-peer.
- Lo stato della connessione peering è attivo.
- Le tabelle dei percorsi delle due VPCs sono impostate correttamente.

Per informazioni sul peering VPC, consulta Utilizzo di connessioni peering VPC.

Client locale

Nel caso di un client locale configurato per connettersi al cluster MSK utilizzando AWS VPN, assicurati quanto segue:

- Lo stato della connessione VPN è UP. Per informazioni su come verificare lo stato della connessione VPN, consulta How do I check the current status of my VPN tunnel?.
- La tabella di routing del VPC del cluster contiene la route per un CIDR locale la cui destinazione ha il formato `Virtual private gateway(vgw-xxxxxxxx)`.
- Il gruppo di sicurezza del cluster MSK consente il traffico sulla porta 2181, sulla porta 9092 (se il cluster accetta traffico non crittografato) e sulla porta 9094 (se il cluster accetta traffico crittografato TLS).

Per ulteriori indicazioni AWS VPN sulla risoluzione dei problemi, consulta Troubleshooting Client VPN.

AWS Direct Connect

Se il client utilizza AWS Direct Connect, consulta Risoluzione dei problemi AWS Direct Connect.

Se le linee guida per risoluzione dei problemi precedenti non consentono di risolvere il problema, assicurarsi che il traffico di rete non sia bloccato da un firewall. Per ulteriori operazioni di debug, utilizza strumenti come `tcpdump` e `Wireshark` per analizzare il traffico e assicurarti che raggiunga il cluster MSK.

Autenticazione non riuscita: troppe connessioni

L'errore `Failed authentication ... Too many connects` indica che un broker si sta proteggendo perché uno o più client IAM stanno tentando di connettersi ad esso a una velocità aggressiva. Per aiutare i broker ad accettare nuove connessioni IAM a una velocità più elevata, puoi aumentare il parametro di configurazione `reconnect.backoff.ms`.

Per ulteriori informazioni sui limiti di velocità per le nuove connessioni per broker, consulta la pagina Quota di Amazon MSK.

Autenticazione fallita: sessione troppo breve

L'`Failed authentication ... Session too short`errore si verifica quando il client tenta di connettersi a un cluster utilizzando credenziali IAM che stanno per scadere. Assicurati di controllare come vengono aggiornate le tue credenziali IAM. Molto probabilmente, le credenziali vengono sostituite troppo vicino alla scadenza della sessione, il che comporta problemi sul lato server e errori di autenticazione.

MSK Serverless: la creazione del cluster ha esito negativo

Se si tenta di creare un cluster MSK Serverless e il flusso di lavoro ha esito negativo, è possibile che non si disponga dell'autorizzazione per creare un endpoint VPC. Verifica che l'amministratore ti abbia concesso l'autorizzazione a creare un endpoint VPC consentendo l'operazione `ec2:CreateVpcEndpoint`.

Per un elenco completo delle autorizzazioni necessarie per eseguire tutte le operazioni di Amazon MSK, consulta la pagina AWS politica gestita: Amazon MSKFull Access.

Impossibile eseguire l'aggiornamento KafkaVersionsList nella configurazione MSK

Quando si aggiorna la KafkaVersionsListproprietà nella AWS::MSK::Configurationrisorsa, l'aggiornamento ha esito negativo e viene visualizzato il seguente errore.

```
Resource of type 'AWS::MSK::Configuration' with identifier '<identifierName>' already
 exists.
```

Quando si aggiorna la `KafkaVersionsList` proprietà, AWS CloudFormation ricrea una nuova configurazione con la proprietà aggiornata prima di eliminare la configurazione precedente. L'aggiornamento AWS CloudFormation dello stack non riesce perché la nuova configurazione utilizza lo stesso nome della configurazione esistente. Tale aggiornamento richiede la sostituzione di una risorsa. Per eseguire correttamente l'aggiornamento`KafkaVersionsList`, è necessario aggiornare anche la proprietà Name nella stessa operazione.

Inoltre, se la configurazione è associata a qualsiasi cluster creato utilizzando AWS Management Console o AWS CLI, aggiungete quanto segue alla risorsa di configurazione per evitare tentativi falliti di eliminazione delle risorse.

```
UpdateReplacePolicy: Retain
```

Una volta completato l'aggiornamento, vai alla console Amazon MSK ed elimina la vecchia configurazione. Per informazioni sulle configurazioni MSK, consulta Configurazione Amazon MSK Provisioned.

Le migliori pratiche per i broker Standard ed Express

Questa sezione descrive le migliori pratiche da seguire per i broker Standard e i broker Express. Per informazioni sulle best practice di Amazon MSK, consulta la pagina. Best practice per l'utilizzo del replicatore MSK

Argomenti

- Best practice per broker standard
- Le migliori pratiche per i broker Express
- Best practice per i client Apache Kafka

Best practice per broker standard

In questo argomento vengono illustrate alcune best practice da seguire quando si utilizza Amazon MSK. Per informazioni sulle best practice di Amazon MSK, consulta la pagina. Best practice per l'utilizzo del replicatore MSK

Considerazioni lato client

La disponibilità e le prestazioni dell'applicazione dipendono non solo dalle impostazioni lato server ma anche dalle impostazioni client.

- Configurazione dei client per l'elevata disponibilità. In un sistema distribuito come Apache Kafka, garantire un'elevata disponibilità è fondamentale per mantenere un'infrastruttura di messaggistica affidabile e tollerante ai guasti. I broker andranno offline per eventi pianificati e non pianificati, ad esempio aggiornamenti, patch, guasti hardware e problemi di rete. Un cluster Kafka è tollerante nei confronti di un broker offline, pertanto i clienti Kafka devono anche gestire il failover dei broker con garbo. Vedi i dettagli completi su. Best practice per i client Apache Kafka
- Assicurati che le stringhe di connessione del client includano almeno un broker per ogni zona di disponibilità. La presenza di più broker nella stringa di connessione di un client consente il failover quando un broker specifico è offline a seguito di un aggiornamento. Per informazioni su come ottenere una stringa di connessione con più broker, consulta Ottieni i broker bootstrap per un cluster Amazon MSK.
- Esegui test delle prestazioni per verificare che le configurazioni dei tuoi client ti consentano di raggiungere i tuoi obiettivi prestazionali.

Considerazioni lato server

Dimensionamento corretto del cluster: numero di partizioni per broker standard

Nella tabella seguente viene illustrato il numero consigliato di partizioni (incluse le repliche leader e follower) per broker Standard. Il numero consigliato di partizioni non viene applicato e rappresenta una procedura ottimale per gli scenari in cui si invia traffico su tutte le partizioni tematiche assegnate.

Dimensionamento del broker	Numero consigliato di partizioni (incluse le repliche leader e follower) per broker	Numero massimo di partizioni che supportano le operazioni di aggiornamento
`kafka.t3.small`	300	300
`kafka.m5.large` o `kafka.m5.xlarge`	1000	1500
`kafka.m5.2xlarge`	2000	3000
`kafka.m5.4xlarge`, `kafka.m5.8xlarge`, `kafka.m5.12xlarge`, `kafka.m5.16xlarge` oppure `kafka.m5.24xlarge`	4000	6000
`kafka.m7g.large` o `kafka.m7g.xlarge`	1000	1500
`kafka.m7g.2xlarge`	2000	3000
`kafka.m7g.4xlarge`, `kafka.m7g.8xlarge` `kafka.m7g.12xlarge`, o `kafka.m7g.16xlarge`	4000	6000

Se si utilizzano partizioni elevate e un throughput ridotto in cui il numero di partizioni è più elevato, ma non si invia traffico su tutte le partizioni, è possibile comprimere più partizioni per broker, purché siano stati eseguiti test e test delle prestazioni sufficienti per verificare che il cluster rimanga integro con un numero di partizioni più elevato. Se il numero di partizioni per broker supera il valore massimo consentito e il cluster si sovraccarica, ti verrà impedito di eseguire le seguenti operazioni:

- Aggiornamento della configurazione del cluster
- Aggiornamento del cluster con una dimensione del broker più piccola
- Associazione di un AWS Secrets Manager segreto a un cluster con autenticazione SASL/SCRAM

Un numero elevato di partizioni può inoltre comportare la mancanza delle metriche di Kafka CloudWatch su e sullo scraping di Prometheus.

Per informazioni sulla scelta del numero di partizioni, consulta Apache Kafka Supports 200K Partitions Per Cluster. Ti consigliamo anche di eseguire test autonomi per determinare la dimensione corretta per i tuoi broker. Per ulteriori informazioni sulle diverse dimensioni dei broker, consultathe section called “Tipi di broker”.

Dimensionamento corretto del cluster: numero di broker standard per cluster

Per determinare il numero corretto di broker Standard per il cluster MSK Provisioned e comprendere i costi, consulta il foglio di calcolo MSK Sizing and Pricing. Questo foglio di calcolo fornisce una stima delle dimensioni di un cluster MSK Provisioned e dei costi associati di Amazon MSK rispetto a un cluster Apache Kafka simile basato sull'autogestimento. EC2 Per ulteriori informazioni sui parametri di input nel foglio di calcolo, passare il mouse sulle descrizioni dei parametri. Le stime fornite da questo foglio sono conservative e forniscono un punto di partenza per un nuovo cluster MSK. Le prestazioni, le dimensioni e i costi del cluster dipendono dal caso d'uso e consigliamo di verificarli con test ad hoc.

Per comprendere in che modo l'infrastruttura sottostante influisce sulle prestazioni di Apache Kafka, consulta l'articolo Best practice for right-sizing your Apache Kafka clusters to optimize performance and cost nel blog sui big data. AWS Il post del blog fornisce informazioni su come dimensionare i cluster per soddisfare i requisiti di velocità di trasmissione effettiva, disponibilità e latenza. Fornisce inoltre risposte a domande quali quando è necessario aumentare o ridurre la capacità e indicazioni su come verificare continuamente le dimensioni dei cluster di produzione. Per informazioni sui cluster basati sullo storage su più livelli, consulta Best practice per l'esecuzione di carichi di lavoro di produzione utilizzando lo storage su più livelli Amazon MSK.

Ottimizzazione della velocità di trasmissione effettiva del cluster per istanze m5.4xl, m7g.4xl o di dimensioni maggiori

Quando si utilizzano istanze m5.4xl, m7g.4xl o di dimensioni maggiori, è possibile ottimizzare la velocità di trasmissione effettiva del cluster MSK Provisioned ottimizzando le configurazioni num.io.threads e num.network.threads.

Il valore num.io.threads è il numero di thread utilizzati da un broker standard per l'elaborazione delle richieste. L'aggiunta di più thread, fino al numero di core CPU supportati per le dimensioni dell'istanza, può contribuire a migliorare la velocità di trasmissione effettiva del cluster.

Il valore num.network.threads è il numero di thread utilizzati dal broker Standard per ricevere tutte le richieste in arrivo e restituire le risposte. I thread di rete inseriscono le richieste in entrata in una coda di richieste per l'elaborazione da parte di io.threads. L'impostazione di num.network.threads sulla metà del numero di core CPU supportati per la dimensione dell'istanza consente l'utilizzo completo della nuova dimensione dell'istanza.

> **Important**
>
> Non aumentare num.network.threads senza prima aumentare num.io.threads, in quanto ciò può causare una congestione legata alla saturazione della coda.

Impostazioni consigliate

Dimensioni istanza	Valore consigliato per num.io.threads	Valore consigliato per num.network.threads
m5.4xl	16	8
m5.8xl	32	16
m5.12xl	48	24
m5.16xl	64	32
m5.24xl	96	48
m7g.4xlarge	16	8
m7g.8xlarge	32	16

Dimensioni istanza	Valore consigliato per num.io.threads	Valore consigliato per num.network.threads
m7g.12xlarge	48	24
m7g.16xlarge	64	32

Utilizzo dell'ultima versione di Kafka AdminClient per evitare problemi di mancata corrispondenza degli ID degli argomenti

L'ID di un argomento viene perso («Error: does not match the topic Id for partition») quando si utilizza una AdminClient versione di Kafka precedente alla 2.8.0 con il flag per aumentare o riassegnare le partizioni degli argomenti per un `--zookeeper` cluster MSK Provisioned utilizzando la versione 2.8.0 o successiva di Kafka. Nota che il flag `--zookeeper` è obsoleto in Kafka 2.5 ed è stato rimosso a partire da Kafka 3.0. Consulta la pagina Upgrading to 2.5.0 from any version 0.8.x through 2.4.x.

Per evitare la mancata corrispondenza degli ID degli argomenti, utilizza una versione del client Kafka 2.8.0 o successiva per le operazioni di amministrazione di Kafka. In alternativa, i client 2.5 e versioni successive possono utilizzare il flag `--bootstrap-servers` al posto del flag `--zookeeper`.

Creazione di cluster a disponibilità elevata

Utilizza i seguenti consigli in modo che i tuoi cluster MSK possano avere un'elevata disponibilità durante un aggiornamento (ad esempio quando aggiorni le dimensioni del broker o la versione di Apache Kafka) o quando Amazon MSK sostituisce un broker.

- Configura un cluster con tre zone di disponibilità.
- Assicurati che il fattore di replica (RF) sia almeno 3. Tieni presente che un valore di RF pari a 1 può portare a partizioni offline durante un aggiornamento in sequenza, mentre un RF pari a 2 può causare la perdita di dati.
- Impostare le repliche in sinc minime (minISR) su al massimo RF - 1. Un minISR uguale a RF può impedire la produzione nel cluster durante un aggiornamento in sequenza. Un minISR di 2 consente di rendere disponibili argomenti replicati a tre vie quando una replica è offline.

Monitoraggio dell'utilizzo della CPU

Amazon MSK consiglia vivamente di mantenere l'utilizzo della CPU per i broker (definito come`CPU User + CPU System`) al di sotto del 60%. Ciò garantisce che il cluster mantenga un margine di

crescita della CPU sufficiente per gestire eventi operativi, come guasti dei broker, applicazione di patch e aggiornamenti continui.

Apache Kafka può ridistribuire il carico della CPU tra i broker del cluster quando necessario. Ad esempio, quando Amazon MSK rileva e ripristina un errore del broker, esegue la manutenzione automatica, ad esempio l'applicazione di patch. Allo stesso modo, quando un utente richiede una modifica delle dimensioni del broker o un aggiornamento di versione, Amazon MSK avvia flussi di lavoro in sequenza che mettono offline un broker alla volta. Quando i broker con partizioni leader vanno offline, Apache Kafka riassegna la leadership delle partizioni per ridistribuire il lavoro agli altri broker del cluster. Seguendo questa best practice, garantisci un margine di crescita della CPU sufficiente per tollerare questi eventi operativi.

> **Note**
>
> Durante il monitoraggio dell'utilizzo della CPU, tenete presente che l'utilizzo totale della CPU include più di e. `CPU User` `CPU System` Anche altre categorie, ad esempio `iowait` `irqsoftirq`, `esteal`, contribuiscono all'attività complessiva della CPU. Di conseguenza, CPU Idle non è sempre uguale a`100% - CPU User - CPU System`.

Puoi utilizzare Amazon CloudWatch Metric Math per creare una metrica composita (`CPU User + CPU System`) e impostare un allarme da attivare quando l'utilizzo medio supera il 60%. Quando viene attivato, valuta la possibilità di dimensionare il cluster utilizzando una delle seguenti opzioni:

- Opzione 1 (consigliata): aggiorna le dimensioni del broker scegliendo la dimensione immediatamente più grande. Ad esempio, se la dimensione corrente è`kafka.m5.large`, aggiorna il cluster da utilizzare`kafka.m5.xlarge`. Tieni presente che quando aggiorni le dimensioni del broker nel cluster, Amazon MSK mette i broker offline progressivamente e riassegna temporaneamente la leadership delle partizioni ad altri broker. Un aggiornamento delle dimensioni richiede in genere 10-15 minuti per broker.
- Opzione 2: se ci sono argomenti in cui tutti i messaggi sono stati acquisiti da produttori che utilizzano scritture ininterrotte (in altre parole, i messaggi non sono codificati e l'ordinamento non è importante per i consumatori), espandi il cluster aggiungendo altri broker. Inoltre, aggiungi partizioni agli argomenti esistenti con la velocità di trasmissione effettiva più elevata. Successivamente, utilizza `kafka-topics.sh --describe` per assicurarti che le partizioni appena aggiunte vengano assegnate ai nuovi broker. Il vantaggio principale di questa opzione rispetto alla precedente è la possibilità di gestire risorse e costi in modo più granulare. Inoltre, è possibile

utilizzare questa opzione se il carico della CPU supera in modo significativo il 60%, poiché questa forma di dimensionamento in genere non comporta un aumento del carico per i broker esistenti.

- Opzione 3: espandi il cluster MSK Provisioned aggiungendo broker, quindi riassegna le partizioni esistenti utilizzando lo strumento di riassegnazione delle partizioni denominato. `kafka-reassign-partitions.sh` Tuttavia, se utilizzi questa opzione, il cluster dovrà spendere risorse per replicare i dati da broker a broker dopo la riassegnazione delle partizioni. Rispetto alle due opzioni precedenti, questa opzione può inizialmente aumentare in modo significativo il carico sul cluster. Di conseguenza, Amazon MSK sconsiglia di utilizzare questa opzione quando l'utilizzo della CPU è superiore al 70%, perché la replica causa un carico aggiuntivo della CPU e del traffico di rete. Amazon MSK consiglia di utilizzare questa opzione solo se le due opzioni precedenti non sono percorribili.

Altre raccomandazioni:

- Monitora l'utilizzo totale della CPU per broker come proxy per la distribuzione del carico. Se i broker hanno un utilizzo della CPU costantemente irregolare, potrebbe essere un segno che il carico non è distribuito uniformemente all'interno del cluster. È consigliabile utilizzare Cruise Control per gestire in modo continuo la distribuzione del carico tramite l'assegnazione delle partizioni.
- Monitora la latenza di produzione e utilizzo. La latenza di produzione e utilizzo può aumentare linearmente con l'utilizzo della CPU.
- Intervallo di scrape JMX: se si abilita il monitoraggio aperto con la funzionalità Prometheus, si consiglia di utilizzare un intervallo di scrape di 60 secondi o superiore (scrape_interval: 60s) per la configurazione dell'host Prometheus (prometheus.yml). La riduzione dell'intervallo di scrape può comportare un utilizzo elevato della CPU sul cluster.

Monitoraggio dello spazio su disco

Per evitare di esaurire lo spazio su disco per i messaggi, crea un CloudWatch allarme che controlla la `KafkaDataLogsDiskUsed` metrica. Quando il valore di questo parametro raggiunge o supera l'85%, esegui una o più delle seguenti operazioni:

- Utilizza the section called “Scalabilità automatica per i cluster”. Puoi anche aumentare manualmente lo spazio di archiviazione del broker come descritto nella sezione the section called “Dimensionamento manuale”.

- Riduci il periodo di conservazione dei messaggi o la dimensione del log. Per informazioni su come eseguire queste operazioni, consulta the section called “Regolazione dei parametri di conservazione dei dati”.
- Elimina argomenti non utilizzati.

Per informazioni su come configurare e utilizzare gli allarmi, consulta Using Amazon CloudWatch Alarms. Per un elenco completo di parametri di Amazon MSK, consulta la sezione the section called “Monitora un cluster”.

Regolazione dei parametri di conservazione dei dati

Il consumo di messaggi non li rimuove dal log. Per liberare regolarmente spazio su disco, puoi specificare in modo esplicito un periodo di conservazione, ovvero il periodo di permanenza dei messaggi nel log. Puoi inoltre specificare una dimensione del log di conservazione. Quando viene raggiunto il periodo di conservazione o la dimensione del log di conservazione, Apache Kafka inizia a rimuovere i segmenti inattivi dal log.

Per specificare una policy di conservazione a livello di cluster, imposta uno o più dei seguenti parametri: `log.retention.hours`, `log.retention.minutes`, `log.retention.ms` o `log.retention.bytes`. Per ulteriori informazioni, consulta the section called “Configurazioni Amazon MSK personalizzate”.

Puoi anche specificare i parametri di conservazione a livello di argomento:

- Per specificare un periodo di conservazione per argomento, utilizza il comando seguente.

```
kafka-configs.sh --bootstrap-server $bs --alter --entity-type topics --entity-
name TopicName --add-config retention.ms=DesiredRetentionTimePeriod
```

- Per specificare una dimensione del log di conservazione per argomento, utilizza il comando seguente.

```
kafka-configs.sh --bootstrap-server $bs --alter --entity-type topics --entity-
name TopicName --add-config retention.bytes=DesiredRetentionLogSize
```

I parametri di conservazione specificati a livello di argomento hanno la precedenza sui parametri a livello di cluster.

Accelerazione del ripristino dei log dopo un arresto non corretto

Dopo un arresto non corretto, un broker può impiegare del tempo per riavviarsi poiché esegue il ripristino dei log. Per impostazione predefinita, Kafka utilizza solo un thread per directory di log per eseguire questo ripristino. Ad esempio, se si dispone di migliaia di partizioni, il completamento del ripristino dei log può richiedere ore. Per velocizzare il ripristino dei log, si consiglia di aumentare il numero di thread utilizzando la proprietà di configurazione `num.recovery.threads.per.data.dir`. È possibile impostarlo sul numero di core CPU.

Monitoraggio della memoria di Apache Kafka

Ti consigliamo di monitorare la memoria utilizzata da Apache Kafka. In caso contrario, il cluster potrebbe diventare non disponibile.

Per determinare la quantità di memoria utilizzata da Apache Kafka, puoi monitorare il parametro `HeapMemoryAfterGC`. `HeapMemoryAfterGC` è la percentuale di memoria heap totale utilizzata dopo la rimozione di oggetti inutili (garbage collection). È consigliabile creare un CloudWatch allarme che si attiva quando `HeapMemoryAfterGC` aumenta oltre il 60%.

Le operazioni che è possibile eseguire per ridurre l'utilizzo della memoria variano. Dipendono dal modo in cui si configura Apache Kafka. Ad esempio, se si utilizza la consegna transazionale dei messaggi, è possibile ridurre il valore `transactional.id.expiration.ms` nella configurazione di Apache Kafka da `604800000` ms a `86400000` ms (da 7 giorni a 1 giorno). Ciò riduce l'ingombro di memoria di ciascuna transazione.

Non aggiungere broker non MSK

Per ZooKeeper i cluster MSK Provisioned, se utilizzi i ZooKeeper comandi Apache per aggiungere broker, questi ultimi non vengono aggiunti al cluster MSK Provisioned e ZooKeeper Apache conterrà informazioni errate sul cluster. Ciò potrebbe comportare la perdita di dati. Per le operazioni supportate del cluster MSK Provisioned, vedere. the section called “Caratteristiche e concetti chiave”

Abilitazione della crittografia dei dati in transito

Per informazioni sulla crittografia dei dati in transito e su come abilitarla, consulta the section called “Crittografia Amazon MSK in transito”.

Riassegnazione delle partizioni

Per spostare le partizioni su broker diversi sullo stesso cluster MSK Provisioned, è possibile utilizzare lo strumento di riassegnazione delle partizioni denominato. `kafka-reassign-partitions.sh` Si consiglia di non riassegnare più di 10 partizioni in una singola chiamata per operazioni sicure.

`kafka-reassign-partitions` Ad esempio, dopo aver aggiunto nuovi broker per espandere un cluster o aver spostato le partizioni per rimuovere i broker, è possibile ribilanciare il cluster riassegnando le partizioni ai nuovi broker. Per informazioni su come aggiungere broker a un cluster MSK, consulta la pagina. the section called “Espandi un cluster” Per informazioni su come rimuovere broker da un cluster MSK, consulta la pagina. the section called “Rimuovi un broker” Per informazioni sullo strumento di riassegnazione delle partizioni, consulta la sezione relativa all'espansione del cluster nella documentazione di Apache Kafka.

Le migliori pratiche per i broker Express

Questo argomento descrive alcune best practice da seguire quando si utilizzano i broker Express. I broker Express sono preconfigurati per garantire disponibilità e durabilità elevate. Per impostazione predefinita, i dati sono distribuiti su tre zone di disponibilità, la replica è sempre impostata su 3 e la replica minima sincronizzata è sempre impostata su 2. Tuttavia, ci sono ancora alcuni fattori da considerare per ottimizzare l'affidabilità e le prestazioni del cluster.

Considerazioni sul lato client

La disponibilità e le prestazioni dell'applicazione dipendono non solo dalle impostazioni lato server ma anche dalle impostazioni client.

- Configura i tuoi client per un'elevata disponibilità. In un sistema distribuito come Apache Kafka, garantire un'elevata disponibilità è fondamentale per mantenere un'infrastruttura di messaggistica affidabile e tollerante ai guasti. I broker andranno offline per eventi pianificati e non pianificati, ad esempio aggiornamenti, patch, guasti hardware e problemi di rete. Un cluster Kafka è tollerante nei confronti di un broker offline, pertanto i clienti Kafka devono anche gestire il failover dei broker con garbo. Consulta i dettagli completi nelle raccomandazioni sulle migliori pratiche per i clienti Apache Kafka.
- Esegui test delle prestazioni per verificare che le configurazioni dei tuoi client ti consentano di raggiungere i tuoi obiettivi prestazionali anche quando riavviamo i broker in condizioni di picco. È possibile riavviare i broker del cluster dalla console MSK o utilizzando MSK. APIs

Considerazioni sul lato server

Argomenti

- Dimensionamento corretto del cluster: numero di broker per cluster
- Monitoraggio dell'utilizzo della CPU

- Dimensioni corrette del cluster: numero di partizioni per broker Express
- Monitora il numero di connessioni
- Riassegnazione delle partizioni

Dimensionamento corretto del cluster: numero di broker per cluster

Scegliere il numero di broker per il cluster basato su Express è semplice. Ogni broker Express è dotato di una capacità di throughput definita per l'ingresso e l'uscita. È consigliabile utilizzare questa capacità di throughput come mezzo principale per dimensionare il cluster (e quindi considerare altri fattori come il numero di partizioni e connessioni, descritti di seguito).

Ad esempio, se la tua applicazione di streaming richiede il 45% MBps della capacità di ingresso (scrittura) e il 90% MBps dei dati in uscita (lettura), puoi semplicemente utilizzare 3 broker express.m7g.large per soddisfare le tue esigenze di throughput. Ogni broker express.m7g.large gestirà il 15% dell'ingresso e il 30% dell'uscita. MBps MBps Consulta la tabella seguente per i limiti di throughput consigliati per ogni dimensione del broker Express. Se il throughput supera i limiti consigliati, potresti riscontrare un peggioramento delle prestazioni e dovresti ridurre il traffico o scalare il cluster. Se la velocità effettiva supera i limiti consigliati e raggiunge la quota per broker, MSK limiterà il traffico dei clienti per evitare ulteriori sovraccarichi.

Puoi anche utilizzare il nostro foglio di calcolo (vedi MSK Sizing and Pricing) per valutare diversi scenari e prendere in considerazione altri fattori, come il numero di partizioni.

Produttività massima consigliata per broker

Dimensioni istanza	Ingresso () MBps	Uscita () MBps
`express.m7g.large`	15.6	31,2
`express.m7g.xlarge`	31,2	62,5
`express.m7g.2xlarge`	62,5	125,0
`express.m7g.4xlarge`	124,9	249,8
`express.m7g.8xlarge`	250,0	500,0
`express.m7g.12xlarge`	375,0	750,0
`express.m7g.16xlarge`	500,0	1000,0

Monitoraggio dell'utilizzo della CPU

Ti consigliamo di mantenere l'utilizzo totale della CPU per i tuoi broker (definito come utente CPU più sistema CPU) al di sotto del 60%. Quando hai a disposizione almeno il 40% della CPU totale del cluster, Apache Kafka può ridistribuire il carico della CPU tra i broker del cluster, se necessario. Ciò può essere necessario a causa di eventi pianificati o non pianificati. Un esempio di evento pianificato è l'aggiornamento di una versione del cluster durante il quale MSK aggiorna i broker di un cluster riavviandoli uno alla volta. Un esempio di evento non pianificato è un guasto hardware in un broker o, nel peggiore dei casi, un guasto AZ in cui tutti i broker di una AZ sono interessati. Quando i broker con partition lead repliche vanno offline, Apache Kafka riassegna la leadership delle partizioni per ridistribuire il lavoro agli altri broker del cluster. Seguendo questa best practice, potete assicurarvi di avere abbastanza spazio di crescita della CPU nel cluster per tollerare eventi operativi come questi.

Puoi usare Using math expression with CloudWatch metrics nella Amazon CloudWatch User Guide per creare una metrica composita che sia CPU User + CPU System. Imposta un allarme che si attiva quando il parametro composito raggiunge un utilizzo medio della CPU del 60%. Quando viene attivato questo allarme, dimensiona il cluster utilizzando una delle seguenti opzioni:

- Opzione 1: aggiorna la dimensione del broker alla dimensione successiva più grande. Tieni presente che quando aggiorni le dimensioni dei broker nel cluster, Amazon MSK disconnette i broker in modo continuativo e riassegna temporaneamente la leadership delle partizioni ad altri broker.
- Opzione 2: espandi il cluster aggiungendo broker, quindi riassegnando le partizioni esistenti utilizzando lo strumento di riassegnazione delle partizioni denominato. `kafka-reassign-partitions.sh`

Altri consigli

- Monitora l'utilizzo totale della CPU per broker come proxy per la distribuzione del carico. Se i broker utilizzano costantemente la CPU in modo disomogeneo, è possibile che il carico non sia distribuito in modo uniforme all'interno del cluster. Consigliamo di utilizzare Cruise Control per gestire continuamente la distribuzione del carico tramite l'assegnazione delle partizioni.
- Monitora la latenza di produzione e utilizzo. La latenza di produzione e utilizzo può aumentare linearmente con l'utilizzo della CPU.
- Intervallo di scrape JMX: se si abilita il monitoraggio aperto con la funzione Prometheus, si consiglia di utilizzare un intervallo di scrape di 60 secondi o superiore () per la configurazione host

Prometheus (). `scrape_interval: 60s` `prometheus.yml` La riduzione dell'intervallo di scrape può comportare un utilizzo elevato della CPU sul cluster.

Dimensioni corrette del cluster: numero di partizioni per broker Express

Se hai un numero elevato di partizioni e un throughput ridotto in cui hai un numero di partizioni più elevato, ma non invii traffico su tutte le partizioni, puoi comprimere più partizioni per broker, purché tu abbia eseguito test e test delle prestazioni sufficienti per verificare che il cluster rimanga integro con un numero di partizioni più elevato. Se il numero di partizioni per broker supera il valore massimo consentito e il cluster si sovraccarica, ti verrà impedito di eseguire le seguenti operazioni:

- Aggiornamento della configurazione del cluster
- Aggiorna il cluster a un broker di dimensioni inferiori
- Associa un AWS Secrets Manager segreto a un cluster con SASL/SCRAM autenticazione

Un cluster sovraccarico con un numero elevato di partizioni può inoltre comportare la mancanza dei parametri di Kafka sullo scraping di Prometheus. CloudWatch

Per informazioni sulla scelta del numero di partizioni, consulta Apache Kafka Supports 200K Partitions Per Cluster. Ti consigliamo inoltre di eseguire i tuoi test per determinare la dimensione giusta per i tuoi broker. Per ulteriori informazioni sulle diverse dimensioni dei broker, consultaDimensioni dei broker Amazon MSK.

Per informazioni sul numero consigliato di partizioni (incluse le repliche leader e follower) per ogni broker Express, consulta. Quota di partizione Express Broker Il numero consigliato di partizioni non viene applicato e rappresenta una procedura consigliata per gli scenari in cui si invia traffico attraverso tutte le partizioni tematiche assegnate.

Monitora il numero di connessioni

Le connessioni dei client ai broker consumano risorse di sistema come memoria e CPU. A seconda del meccanismo di autenticazione utilizzato, è necessario effettuare un monitoraggio per assicurarsi di rispettare i limiti applicabili. Per gestire i tentativi di connessione non riusciti, puoi impostare il parametro di configurazione `reconnect.backoff.ms` sul lato client. Ad esempio, se desideri che un client riprovi a connettersi dopo 1 secondo, imposta `sureconnect.backoff.ms. 1000` Per ulteriori informazioni sulla configurazione dei nuovi tentativi, consulta la documentazione di Apache Kafka.

Dimensione	Quota
Numero massimo di connessioni TCP per broker (controllo degli accessi IAM)	3000
Numero massimo di connessioni TCP per broker (IAM)	100 al secondo
Numero massimo di connessioni TCP per broker (non IAM)	MSK non impone limiti di connessione per l'autenticazione non IAM. È tuttavia necessari o monitorare altre metriche come l'utilizzo della CPU e della memoria per assicurarsi di non sovraccaricare il cluster a causa di connessioni eccessive.

Riassegnazione delle partizioni

Per spostare le partizioni su broker diversi sullo stesso cluster MSK Provisioned, è possibile utilizzare lo strumento di riassegnazione delle partizioni denominato. `kafka-reassign-partitions.sh` Si consiglia di non riassegnare più di 20 partizioni in una singola chiamata per operazioni sicure. `kafka-reassign-partitions` Ad esempio, dopo aver aggiunto nuovi broker per espandere un cluster o aver spostato le partizioni per rimuovere i broker, è possibile ribilanciare il cluster riassegnando le partizioni ai nuovi broker. Per informazioni su come aggiungere broker a un cluster MSK Provisioned, vedere. the section called “Espandi un cluster” Per informazioni su come rimuovere broker da un cluster MSK Provisioned, vedere. the section called “Rimuovi un broker” Per informazioni sullo strumento di riassegnazione delle partizioni, consulta la sezione relativa all'espansione del cluster nella documentazione di Apache Kafka.

Best practice per i client Apache Kafka

Quando si lavora con Apache Kafka e Amazon MSK, è importante configurare correttamente sia il client che il server per prestazioni e affidabilità ottimali. Questa guida fornisce consigli sulla configurazione lato client basata sulle best practice per Amazon MSK.

Per ulteriori informazioni sulle best practice di Amazon MSK Replicator, consulta la sezione. Best practice per l'utilizzo del replicatore MSK Per le best practice dei broker Standard ed Express, consulta. Le migliori pratiche per i broker Standard ed Express

Argomenti

Disponibilità del client Apache Kafka

In un sistema distribuito come Apache Kafka, garantire un'elevata disponibilità è fondamentale per mantenere un'infrastruttura di messaggistica affidabile e tollerante ai guasti. I broker andranno offline per eventi pianificati e non pianificati, come aggiornamenti, patch, guasti hardware e problemi di rete. Un cluster Kafka è tollerante nei confronti di un broker offline, pertanto i clienti Kafka devono anche gestire il failover dei broker con garbo. Per garantire un'elevata disponibilità dei clienti Kafka, consigliamo queste best practice.

Disponibilità producer

- Impostato `retries` per indicare al produttore di riprovare a inviare messaggi non riusciti durante il failover del broker. Consigliamo un valore intero massimo o un valore elevato simile per la maggior parte dei casi d'uso. In caso contrario, si interromperà l'elevata disponibilità di Kafka.
- Imposta `delivery.timeout.ms` per specificare il limite superiore per il tempo totale tra l'invio di un messaggio e la ricezione di una conferma dal broker. Ciò dovrebbe riflettere i requisiti aziendali relativi alla durata di validità di un messaggio. Imposta il limite di tempo sufficientemente alto da consentire un numero sufficiente di tentativi per completare l'operazione di failover. Si consiglia un valore pari o superiore a 60 secondi per la maggior parte dei casi d'uso.
- Impostato `request.timeout.ms` al valore massimo, una singola richiesta deve attendere prima di tentare un nuovo invio. Consigliamo un valore pari o superiore a 10 secondi per la maggior parte dei casi d'uso.
- Imposta `retry.backoff.ms` per configurare il ritardo tra i tentativi per evitare tempeste di tentativi e impatto sulla disponibilità. Consigliamo un valore minimo di 200 ms per la maggior parte dei casi d'uso.
- Imposta `acks=all` per configurare una durabilità elevata; questa impostazione deve essere in linea con una configurazione lato server di `RF=3` e `min.isr=2` garantire che tutte le partizioni in ISR riconoscano la scrittura. Durante un singolo broker offline, questo è il, cioè. `min.isr` 2

Disponibilità del consumer

- Impostata `latest` inizialmente `auto.offset.reset` per gruppi di consumatori nuovi o ricreati. In questo modo si evita il rischio di aggiungere carico al cluster consumando l'intero argomento.
- Impostato `auto.commit.interval.ms` quando si utilizza`enable.auto.commit`. Si consiglia un valore minimo di 5 secondi per la maggior parte dei casi d'uso per evitare il rischio di carico aggiuntivo.
- Implementa la gestione delle eccezioni all'interno del codice di elaborazione dei messaggi del consumatore per gestire errori transitori, ad esempio un interruttore automatico o una sospensione con back-off esponenziale. In caso contrario, è possibile che venga generato come errore dell'applicazione, che potrebbe causare un riequilibrio eccessivo.
- Imposta `isolation.level` per controllare la modalità di lettura dei messaggi transazionali:

  Per impostazione predefinita, consigliamo di impostare sempre in `read_uncommitted` modo implicito. Questo non è presente in alcune implementazioni del client.

  Quando si utilizza lo storage su più livelli, `read_uncommitted` si consiglia un valore pari a
- `client.rack`Impostare per utilizzare la lettura della replica più vicina. Si consiglia di impostare su per ridurre `az id` al minimo i costi e la latenza del traffico di rete. Consulta Riduci i costi del traffico di rete dei tuoi utenti Amazon MSK grazie alla consapevolezza dei rack.

Nuovi equilibri del consumer

- Imposta su `session.timeout.ms` un valore maggiore del tempo di avvio di un'applicazione, incluso qualsiasi jitter di avvio implementato. Consigliamo di utilizzare il valore di 60 secondi per la maggior parte dei casi d'uso.
- Imposta `heartbeat.interval.ms` per perfezionare il modo in cui il coordinatore del gruppo considera un consumatore sano. Consigliamo di utilizzare il valore di 10 secondi per la maggior parte dei casi d'uso.
- Imposta uno shutdown hook nell'applicazione per chiudere in modo sicuro il consumatore su SIGTERM, anziché affidarti ai timeout delle sessioni per identificare quando un consumatore lascia un gruppo. Le applicazioni Kstream possono impostare un valore di. `internal.leave.group.on.close` `true`
- Impostato `group.instance.id` su un valore distinto all'interno del gruppo di consumatori. Idealmente un nome host, un task-id o un pod-id. Ti consigliamo di impostarlo sempre per

comportamenti più deterministici e una migliore correlazione tra log client/server durante la risoluzione dei problemi.

- Impostato su `group.initial.rebalance.delay.ms` un valore in linea con il tempo medio di implementazione. In questo modo si interrompono i ribilanciamenti continui durante l'implementazione.
- Impostato `partition.assignment.strategy` per utilizzare gli assegnatori permanenti. Consigliamo l'uno o l'altro. `StickyAssignor` `CooperativeStickyAssignor`

Prestazioni del client Apache Kafka

Per garantire prestazioni elevate dei clienti Kafka, consigliamo queste best practice.

Prestazioni producer

- Impostato `linger.ms` per controllare la quantità di tempo che un produttore attende per il riempimento di un batch. I batch più piccoli sono costosi dal punto di vista computazionale per Kafka in quanto si traducono in più thread e operazioni di I/O contemporaneamente. Consigliamo di utilizzare la seguente impostazione:

  Un valore minimo di 5 ms per tutti i casi d'uso, inclusa la bassa latenza.

  Consigliamo un valore più alto di 25 ms, per la maggior parte dei casi d'uso.

  Si consiglia di non utilizzare mai un valore pari a zero in casi d'uso a bassa latenza. (Un valore pari a zero in genere causa latenza indipendentemente dal sovraccarico di I/O).
- Impostato `batch.size` per controllare la dimensione del batch inviato al cluster. Si consiglia di aumentarlo a un valore di 64 KB o 128 KB.
- Impostato `buffer.memory` quando si utilizzano batch di dimensioni maggiori. Consigliamo di utilizzare un valore di 64 MB per la maggior parte dei casi d'uso.
- Impostato `send.buffer.bytes` per controllare il buffer TCP utilizzato per ricevere i byte. Consigliamo un valore pari a -1 per consentire al sistema operativo di gestire questo buffer quando si esegue un produttore su una rete ad alta latenza.
- Imposta compression.type per controllare la compressione dei batch. Consigliamo lz4 o zstd di eseguire un produttore su una rete ad alta latenza.

Prestazioni del consumer

- Impostato `fetch.min.bytes` per controllare la dimensione minima di recupero valida per ridurre il numero di recuperi e il carico del cluster.

  Consigliamo un valore minimo di 32 byte per tutti i casi d'uso.

  Si consiglia un valore più alto di 128 byte per la maggior parte dei casi d'uso.
- Imposta fetch.max.wait.ms per determinare per quanto tempo il consumatore aspetterà prima che fetch.min.bytes venga ignorato. Consigliamo di utilizzare un valore di 1000 ms per la maggior parte dei casi d'uso.
- Consigliamo che il numero di consumer sia almeno uguale al numero di partizioni per migliorare il parallelismo e la resilienza. In alcune situazioni, potresti decidere di avere meno consumer rispetto al numero di partizioni per argomenti con throughput ridotto.
- Impostato `receive.buffer.bytes` per controllare il buffer TCP utilizzato per ricevere i byte. Consigliamo un valore pari a -1 per consentire al sistema operativo di gestire questo buffer quando si esegue un consumer su una rete ad alta latenza.

Connessioni client

Il ciclo di vita delle connessioni ha un costo computazionale e di memoria su un cluster Kafka. Troppe connessioni create contemporaneamente causano un carico che può influire sulla disponibilità di un cluster Kafka. Questo impatto sulla disponibilità può spesso portare le applicazioni a creare ancora più connessioni, causando così un errore a cascata, con conseguente interruzione completa. È possibile ottenere un numero elevato di connessioni se create a una velocità ragionevole.

Consigliamo le seguenti attenuazioni per gestire alti tassi di creazione di connessioni:

- Assicurati che il meccanismo di distribuzione delle applicazioni non riavvii tutti i produttori/consumatori contemporaneamente, ma preferibilmente in batch più piccoli.
- A livello di applicazione, lo sviluppatore deve assicurarsi che venga eseguito un jitter casuale (random sleep) prima di creare un client di amministrazione, un client di produzione o un client consumer.
- A SIGTERM, quando si chiude la connessione, è necessario eseguire uno sleep casuale per garantire che non tutti i client Kafka vengano chiusi contemporaneamente. Il sonno casuale deve avvenire entro il timeout precedente al verificarsi di SIGKILL.

Example Esempio A (Java)

```
sleepInSeconds(randomNumberBetweenOneAndX);
                        this.kafkaProducer = new KafkaProducer<>(this.props);
```

Example Esempio B (Java)

```
Runtime.getRuntime().addShutdownHook(new Thread(() -> {
    sleepInSeconds(randomNumberBetweenOneAndTwentyFive);
    kafkaProducer.close(Duration.ofSeconds(5));
});
```

- A livello di applicazione, lo sviluppatore deve assicurarsi che i client vengano creati una sola volta per applicazione secondo uno schema singleton. Ad esempio, quando si utilizza lambda, il client deve essere creato in un ambito globale e non nel gestore del metodo.
- Consigliamo di monitorare il numero di connessioni con l'obiettivo di rimanere stabile. creation/close/shiftLa connessione è normale durante le implementazioni e il failover del broker.

Monitoraggio client Kafka

Il monitoraggio dei clienti Kafka è fondamentale per mantenere la salute e l'efficienza del vostro ecosistema Kafka. Che tu sia un amministratore di Kafka, uno sviluppatore o un membro del team operativo, abilitare le metriche lato client è fondamentale per comprendere l'impatto aziendale durante eventi pianificati e non pianificati.

Ti consigliamo di monitorare le seguenti metriche lato client utilizzando il tuo meccanismo di acquisizione delle metriche preferito.

Quando richiedi assistenza con AWS, includi eventuali valori anomali osservati durante l'incidente. Includi anche un esempio dei log dell'applicazione client che descrivono in dettaglio gli errori (non gli avvisi).

Metriche producer

- byte-rate
- record-send-rate
- records-per-request-avg

- acks-latency-avg
- request-latency-avg
- request-latency-max
- record-error-rate
- record-retry-rate
- tasso di errore

> **Note**
>
> Gli errori transitori relativi ai nuovi tentativi non sono motivo di preoccupazione, in quanto fanno parte del protocollo di Kafka per la gestione di problemi transitori come il failover dei leader o le ritrasmissioni di rete. `record-send-rate`confermerà se i produttori stanno ancora procedendo con i nuovi tentativi.

Metriche del consumer

- records-consumed-rate
- bytes-consumed-rate
- frequenza di recupero
- records-lag-max
- record-error-rate
- fetch-error-rate
- polling rate
- rebalance-latency-avg
- tasso di commissione

> **Note**
>
> Frequenze di fetch e commitrate elevate causeranno un carico non necessario sul cluster. È ottimale eseguire richieste in batch più grandi.

Metriche comuni

- connection-close-rate
- connection-creation-rate
- conteggio connessioni

> **Note**
>
> La creazione/terminazione di una connessione elevata causerà un carico non necessario sul cluster.

Cos'è MSK Serverless?

> **Note**
>
> MSK Serverless è disponibile nelle regioni Stati Uniti orientali (Ohio), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Canada (Centrale), Asia Pacifico (Mumbai), Asia Pacifico (Singapore), Asia Pacifico (Sydney), Asia Pacifico (Tokyo), Asia Pacifico (Seoul), Europa (Francoforte), Europa (Stoccolma), Europa (Irlanda), Europa (Parigi) ed Europa (Londra).

MSK Serverless è un tipo di cluster per Amazon MSK che consente di eseguire Apache Kafka senza dover gestire e dimensionare la capacità del cluster. Fornisce e dimensiona automaticamente la capacità durante la gestione delle partizioni dell'argomento, in modo da poter trasmettere i dati senza pensare all'adeguamento o al dimensionamento dei cluster. MSK Serverless offre un modello tariffario basato sulla velocità di trasmissione effettiva, perciò ti viene addebitato soltanto l'utilizzo effettivo. Se le tue applicazioni richiedono una capacità di streaming on demand con aumento e riduzione automatiche, prendi in considerazione l'utilizzo di un cluster serverless.

MSK Serverless è completamente compatibile con Apache Kafka, quindi è possibile utilizzare qualsiasi applicazione client compatibile per produrre e utilizzare dati. Inoltre, si integra con i seguenti servizi:

- AWS PrivateLink per fornire connettività privata
- AWS Identity and Access Management (IAM) per l'autenticazione e l'autorizzazione utilizzando linguaggi Java e non Java. Per istruzioni sulla configurazione dei client per IAM, consulta Configurazione dei client per il Controllo degli accessi IAM.
- AWS Glue Registro degli schemi per la gestione degli schemi
- Servizio gestito da Amazon per Apache Flink per l'elaborazione di flussi basata su Apache Flink
- AWS Lambda per l'elaborazione degli eventi

> **Note**
>
> MSK Serverless richiede il Controllo degli accessi IAM per tutti i cluster. Le liste di controllo degli accessi di Apache Kafka (ACLs) non sono supportate. Per ulteriori informazioni, consulta the section called “Controllo degli accessi IAM”.

Per informazioni sulle quote di servizio applicabili a MSK Serverless, consulta la sezione the section called “Quota per i cluster serverless”.

Per iniziare a utilizzare i cluster serverless e per ulteriori informazioni sulle opzioni di configurazione e monitoraggio per i cluster serverless, consulta le seguenti risorse.

Argomenti

- Utilizzate i cluster MSK Serverless
- Proprietà di configurazione per i cluster MSK Serverless
- Monitora i cluster MSK Serverless

Utilizzate i cluster MSK Serverless

Questo tutorial mostra un esempio di come creare un cluster MSK Serverless, creare un computer client in grado di accedervi e utilizzare il client per creare argomenti sul cluster e scrivere dati su tali argomenti. Questo esempio non rappresenta tutte le opzioni che è possibile scegliere quando si crea un cluster serverless. In diverse parti di questo esercizio verranno scelte opzioni predefinite per semplicità. Ciò non significa che siano le uniche opzioni che funzionano per la configurazione del cluster serverless. Puoi anche utilizzare l'API AWS CLI o Amazon MSK. Per ulteriori informazioni, consulta la documentazione di riferimento all'API di Amazon MSK 2.0.

Argomenti

- Crea un cluster MSK Serverless
- Crea un ruolo IAM per gli argomenti sul cluster MSK Serverless
- Crea una macchina client per accedere al cluster MSK Serverless
- Crea un argomento di Apache Kafka
- Produci e consuma dati in MSK Serverless
- Eliminare le risorse create per MSK Serverless

Crea un cluster MSK Serverless

In questo passaggio, eseguirai due attività. Innanzitutto, si crea un cluster MSK Serverless con le impostazioni predefinite. In secondo luogo, si raccolgono informazioni sul cluster. Si tratta di

informazioni che ti occorreranno nei passaggi successivi, quando creerai un client in grado di inviare dati al cluster.

Creazione di un cluster serverless

1. Accedi a e apri la console Amazon MSK da https://console.aws.amazon.com/msk/casa. AWS Management Console
2. Scegli Create cluster (Crea cluster).
3. Per Metodo di creazione, lascia selezionata l'opzione Creazione rapida. L'opzione Creazione rapida consente di creare un cluster serverless con le impostazioni predefinite.
4. In Nome del cluster, inserisci un nome descrittivo, ad esempio **`msk-serverless-tutorial-cluster`**.
5. In Proprietà generali del cluster, scegli Serverless come Tipo di cluster. Utilizza i valori predefiniti per le Proprietà generali del cluster rimanenti.
6. Nota la tabella in Tutte le impostazioni del cluster. Questa tabella elenca i valori predefiniti per impostazioni importanti come rete e disponibilità e indica se è possibile modificare ogni impostazione dopo aver creato il cluster. Per modificare un'impostazione prima di creare il cluster, è necessario scegliere l'opzione Creazione personalizzata in Metodo di creazione.

   > **Note**
   >
   > Puoi connettere client da un massimo di cinque diversi VPCs con i cluster MSK Serverless. Per aiutare le applicazioni client a passare a un'altra zona di disponibilità in caso di interruzione, è necessario specificare almeno due sottoreti in ogni VPC.

7. Scegli Create cluster (Crea cluster).

Raccolta delle informazioni sul cluster

1. Nella pagina Riepilogo del cluster, scegli Visualizza informazioni sul client. Questo pulsante rimane disattivato fino al termine della creazione del cluster da parte di Amazon MSK. Potrebbe essere necessario attendere qualche minuto prima che il pulsante diventi attivo e possa essere selezionato.
2. Copia la stringa sotto l'etichetta Endpoint. Questa è la stringa del tuo server di bootstrap.
3. Scegliere la scheda Properties (Proprietà).

4. Nella sezione Impostazioni di rete, copia le sottoreti e il gruppo IDs di sicurezza e salvali perché queste informazioni saranno necessarie in seguito per creare una macchina client.
5. Scegli una delle sottoreti. Si apre la console Amazon VPC. Cerca l'ID dell'Amazon VPC associato al VPC della sottorete. Salva questo ID dell'Amazon VPC per utilizzarlo in futuro.

Fase successiva

Crea un ruolo IAM per gli argomenti sul cluster MSK Serverless

Crea un ruolo IAM per gli argomenti sul cluster MSK Serverless

In questo passaggio, eseguirai due attività. La prima attività consiste nel creare una policy IAM che consenta l'accesso alla creazione di argomenti nel cluster e all'invio di dati a tali argomenti. La seconda attività consiste nel creare un ruolo IAM e associarvi questa policy. In un passaggio successivo, si crea un computer client che assume questo ruolo e lo utilizza per creare un argomento nel cluster e per inviare dati a quell'argomento.

Creazione di una policy IAM che consenta di creare argomenti e scrivere su di essi

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.
3. Scegliere Create Policy (Crea policy).
4. Scegli la scheda JSON, quindi sostituisci il JSON nella finestra dell'editor con il seguente JSON.

   Nell'esempio seguente, sostituisci quanto segue:

   - *region*con il codice del Regione AWS luogo in cui hai creato il cluster.
   - Esempio di ID dell'account*123456789012*, con il tuo Account AWS ID.
   - *msk-serverless-tutorial-cluster*/*c07c74ea-5146-4a03-add1-9baa787a5b14-s3*e *msk-serverless-tutorial-cluster* con l'ID del cluster serverless e il nome dell'argomento.

   JSON

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:DescribeCluster"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:123456789012:cluster/msk-serverless-
tutorial-cluster/c07c74ea-5146-4a03-add1-9baa787a5b14-s3"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:CreateTopic",
                "kafka-cluster:WriteData",
                "kafka-cluster:DescribeTopic"
            ],
            "Resource": [
            "arn:aws:kafka:us-east-1:123456789012:topic/msk-serverless-
tutorial-cluster/*"
            ]
        }
    ]
}
```

Per istruzioni su come scrivere policy sicure, vedithe section called “Controllo degli accessi IAM”.

5. Scegliere Next: Tags (Successivo: Tag).
6. Scegliere Next:Review (Successivo: Rivedi).
7. Per il nome della policy, inserisci un nome descrittivo, ad esempio **`msk-serverless-tutorial-policy`**.
8. Scegliere Create Policy (Crea policy).

Creazione di un ruolo IAM e collegamento della policy al ruolo

1. Nel riquadro di navigazione, seleziona Ruoli.
2. Scegliere Crea ruolo.
3. In Casi d'uso comuni, scegli EC2, quindi scegli Avanti: Autorizzazioni.

4. Nella casella di ricerca, inserisci il nome della policy creata in precedenza per questo tutorial. Seleziona quindi la casella a sinistra della policy.
5. Scegliere Next: Tags (Successivo: Tag).
6. Scegliere Next:Review (Successivo: Rivedi).
7. Per il nome del ruolo, inserisci un nome descrittivo, ad esempio **`msk-serverless-tutorial-role`**.
8. Scegliere Crea ruolo.

Fase successiva

Crea una macchina client per accedere al cluster MSK Serverless

Crea una macchina client per accedere al cluster MSK Serverless

In questo passaggio, eseguirai due attività. La prima operazione consiste nel creare un' EC2 istanza Amazon da utilizzare come macchina client Apache Kafka. La seconda attività consiste nell'installare gli strumenti Java e Apache Kafka sul computer.

Per creare un computer client

1. Apri la EC2 console Amazon all'indirizzo https://console.aws.amazon.com/ec2/.
2. Scegliere Launch Instance (Avvia istanza).
3. Inserisci un Nome descrittivo per il computer client, ad esempio **`msk-serverless-tutorial-client`**.
4. Lascia Amazon Linux 2 AMI (HVM) - Kernel 5.10, tipo di volume SSD selezionato per Tipo di Amazon Machine Image (AMI).
5. Lascia selezionato il tipo di istanza t2.micro.
6. In Coppia di chiavi (accesso), scegli Crea una nuova coppia di chiavi. Inserisci **`MSKServerlessKeyPair`** per Nome coppia di chiavi. Quindi scegli Scarica coppia di chiavi. In alternativa, è possibile utilizzare una coppia di chiavi esistente.
7. Per Impostazioni di rete, scegli Modifica.
8. In VPC, inserisci l'ID del cloud privato virtuale (VPC) per il cluster serverless. Si tratta del VPC basato sul servizio Amazon VPC il cui ID è stato salvato dopo la creazione del cluster.
9. Per Sottorete, scegli la sottorete di cui hai salvato l'ID dopo aver creato il cluster.

10. Per Firewall (gruppi di sicurezza), seleziona il gruppo di sicurezza associato al cluster. Questo valore funziona se il gruppo di sicurezza ha una regola in entrata che consente il traffico dal gruppo di sicurezza verso sé stesso. Con questa regola, i membri dello stesso gruppo di sicurezza possono comunicare tra loro. Per ulteriori informazioni, consulta la pagina Security group rules nella Guida per gli sviluppatori di Amazon VPC.
11. Espandi la sezione Dettagli avanzati e scegli il ruolo IAM che hai creato nel Crea un ruolo IAM per gli argomenti sul cluster MSK Serverless.
12. Scegli Avvia.
13. Nel riquadro di navigazione a sinistra, scegliere Istanze. Quindi scegli la casella di controllo nella riga che rappresenta l' EC2istanza Amazon appena creata. D'ora in avanti, chiameremo questa istanza computer client.
14. Scegli Connetti e segui le istruzioni per connetterti al computer client.

Configurazione degli strumenti client Apache Kafka sul computer client

1. Per installare Java, esegui il comando seguente sul computer client:

```
sudo yum -y install java-11
```

2. Per recuperare gli strumenti di Apache Kafka necessari per creare argomenti e inviare dati, esegui i seguenti comandi:

```
wget https://archive.apache.org/dist/kafka/2.8.1/kafka_2.12-2.8.1.tgz
```

```
tar -xzf kafka_2.12-2.8.1.tgz
```

> **Note**
>
> Dopo aver estratto l'archivio Kafka, assicurati che gli script nella `bin` directory abbiano i permessi di esecuzione appropriati. Per farlo, esegui il comando seguente.
>
> ```
> chmod +x kafka_2.12-2.8.1/bin/*.sh
> ```

3. Vai alla directory `kafka_2.12-2.8.1/libs`, quindi esegui il comando per scaricare il file JAR IAM di Amazon MSK. Il file JAR IAM di Amazon MSK consente al computer client di accedere al cluster.

```
wget https://github.com/aws/aws-msk-iam-auth/releases/download/v2.3.0/aws-msk-iam-auth-2.3.0-all.jar
```

Utilizzando questo comando, puoi anche scaricare versioni diverse o più recenti del file JAR Amazon MSK IAM.

4. Vai alla directory `kafka_2.12-2.8.1/bin`. Copia le impostazioni delle proprietà seguenti e incollale in un nuovo file. Assegna al file il nome `client.properties` e salvalo.

```
security.protocol=SASL_SSL
sasl.mechanism=AWS_MSK_IAM
sasl.jaas.config=software.amazon.msk.auth.iam.IAMLoginModule required;
sasl.client.callback.handler.class=software.amazon.msk.auth.iam.IAMClientCallbackHandler
```

Fase successiva

Crea un argomento di Apache Kafka

Crea un argomento di Apache Kafka

In questo passaggio, si utilizza il computer client creato in precedenza per creare un argomento sul cluster serverless.

Argomenti

- Configurazione dell'ambiente per la creazione di argomenti
- Creazione di un argomento e scrittura di dati su di esso

Configurazione dell'ambiente per la creazione di argomenti

- Prima di creare un argomento, assicuratevi di aver scaricato il file JAR AWS MSK IAM nella directory di installazione di Kafka. `libs/` Se non l'hai ancora fatto, esegui il seguente comando nella directory di Kafka. `libs/`

```
wget https://github.com/aws/aws-msk-iam-auth/releases/download/v2.3.0/aws-msk-iam-auth-2.3.0-all.jar
```

Questo file JAR è necessario per l'autenticazione IAM con il cluster MSK Serverless.

- Quando si eseguono i comandi Kafka, potrebbe essere necessario assicurarsi che `classpath` includano il file JAR AWS MSK IAM. Per ottenere ciò, procedi in uno dei seguenti modi:
  - Imposta la variabile di `CLASSPATH` ambiente per includere le tue librerie Kafka, come mostrato nell'esempio seguente.

```
export CLASSPATH=<path-to-your-kafka-installation>/libs/*:<path-to-your-kafka-installation>/libs/aws-msk-iam-auth-2.3.0-all.jar
```

  - Esegui i comandi Kafka utilizzando il comando Java completo con explicit`classpath`, come mostrato nell'esempio seguente.

```
java -cp "<path-to-your-kafka-installation>/libs/*:<path-to-your-kafka-installation>/libs/aws-msk-iam-auth-2.3.0-all.jar"
 org.apache.kafka.tools.TopicCommand --bootstrap-server $BS --command-config
 client.properties --create --topic msk-serverless-tutorial --partitions 6
```

Creazione di un argomento e scrittura di dati su di esso

1. Nel `export` comando seguente, sostituisci *`my-endpoint`* con la stringa bootstrap-server che hai salvato dopo aver creato il cluster. Quindi, vai alla directory `kafka_2.12-2.8.1/bin` sul computer client ed esegui il comando `export`.

```
export BS=my-endpoint
```

2. Esegui il comando seguente per creare un argomento chiamato `msk-serverless-tutorial`.

```
<path-to-your-kafka-installation>/bin/kafka-topics.sh --bootstrap-server $BS
 --command-config client.properties --create --topic msk-serverless-tutorial --
partitions 6
```

Fase successiva

Produci e consuma dati in MSK Serverless

Produci e consuma dati in MSK Serverless

In questo passaggio, si producono e si utilizzano dati utilizzando l'argomento creato nel passaggio precedente.

Per produrre e consumare messaggi

1. Esegui il comando seguente per creare un produttore della console.

```
<path-to-your-kafka-installation>/bin/kafka-console-producer.sh --broker-list $BS
 --producer.config client.properties --topic msk-serverless-tutorial
```

2. Immettere qualsiasi messaggio desiderato e premere Enter (Invio). Ripetere questa fase due o tre volte. Ogni volta che immetti una riga e premi Invio, tale riga viene inviata al cluster Apache Kafka come un messaggio separato.
3. Mantenere aperta la connessione al computer client, quindi aprire una seconda connessione separata al computer in una nuova finestra.
4. Utilizza la tua seconda connessione al computer client per creare un utente della console eseguendo il comando seguente. Sostituisci *my-endpoint* con la stringa del server di bootstrap che hai salvato dopo aver creato il cluster.

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server my-endpoint --consumer.config client.properties --topic msk-serverless-
tutorial --from-beginning
```

   Si iniziano a vedere i messaggi immessi in precedenza quando è stato utilizzato il comando produttore della console.
5. Immettere altri messaggi nella finestra del produttore e guardali apparire nella finestra del consumatore.

Se riscontri `classpath` problemi durante l'esecuzione di questi comandi, assicurati di eseguirli dalla directory corretta. Inoltre, assicuratevi che il file JAR AWS MSK IAM sia nella `libs` directory. In alternativa, è possibile eseguire i comandi Kafka utilizzando il comando Java completo con explicit`classpath`, come illustrato nell'esempio seguente.

```
java -cp "kafka_2.12-2.8.1/libs/*:kafka_2.12-2.8.1/libs/aws-msk-iam-auth-2.3.0-
all.jar" org.apache.kafka.tools.ConsoleProducer –broker-list $BS –producer.config
 client.properties –topic msk-serverless-tutorial
```

Fase successiva

Eliminare le risorse create per MSK Serverless

Eliminare le risorse create per MSK Serverless

In questo passaggio, elimini le risorse che hai creato in questo tutorial.

Eliminazione del cluster

1. Apri la console Amazon MSK a https://console.aws.amazon.com/msk/casa.
2. Nell'elenco dei cluster, scegli il cluster che hai creato per questo tutorial.
3. In Operazioni, scegli Elimina cluster.
4. Inserisci `delete` nel campo e scegli Elimina.

Arresto del computer client

1. Apri la EC2 console Amazon all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nell'elenco delle EC2 istanze Amazon, scegli la macchina client che hai creato per questo tutorial.
3. Scegli Stato istanza, quindi scegli Termina istanza.
4. Scegliere Terminate (Termina).

Eliminazione del ruolo e della policy IAM

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Ruoli.
3. Nella casella di ricerca, inserisci il nome del ruolo IAM creato per questo tutorial.
4. Seleziona il ruolo. Quindi scegli Elimina ruolo e conferma l'eliminazione.
5. Nel riquadro di navigazione, seleziona Policy.
6. Nella casella di ricerca, inserisci il nome della policy creata per questo tutorial.
7. Scegli la policy per aprirne la pagina di riepilogo. Nella pagina di Riepilogo della policy, seleziona Elimina policy.
8. Scegliere Delete (Elimina).

Proprietà di configurazione per i cluster MSK Serverless

Amazon MSK imposta le proprietà di configurazione del broker per i cluster serverless. Non è possibile modificare queste impostazioni delle proprietà di configurazione del broker. Tuttavia, è possibile impostare o modificare le seguenti proprietà di configurazione a livello di argomento. Tutte le altre proprietà di configurazione a livello di argomento non sono configurabili.

Proprietà di configurazione	Predefinita	Modificabile	Valore massimo consentito
cleanup.policy	Eliminazione	Sì, ma solo al momento della creazione dell'argomento	
compression.type	Producer	Sì	
max.message.bytes	1048588	Sì	8388608 (8 MiB)
message.timestamp.difference.max.ms	long.max	Sì	
message.timestamp.type	CreateTime	Sì	
retention.bytes	250 GiB	Sì	Illimitato; impostalo su -1 per una conservazione illimitata
retention.ms	7 giorni	Sì	Illimitato; impostalo su -1 per una conservazione illimitata

Per impostare o modificare queste proprietà di configurazione a livello di argomento, puoi utilizzare gli strumenti da riga di comando di Apache Kafka. Vedi 3.2 Configurazioni a livello di argomento nella documentazione ufficiale di Apache Kafka per ulteriori informazioni ed esempi su come impostarle.

> **Note**
>
> Non è possibile modificare la configurazione segment.bytes per gli argomenti in MSK Serverless. Tuttavia, un'applicazione Kafka Streams potrebbe tentare di creare un argomento interno con un valore di configurazione segment.bytes, che è diverso da quello consentito da MSK Serverless. Per informazioni sulla configurazione di Kafka Streams con MSK Serverless, vedere. Utilizzo di Kafka Streams con i broker MSK Express e MSK Serverless

Quando utilizzi gli strumenti da riga di comando di Apache Kafka con Amazon MSK Serverless, assicurati di aver completato i passaggi 1-4 nella sezione Configurare gli strumenti client Apache Kafka sulla macchina client della documentazione Getting Started di Amazon MSK Serverless. Inoltre, è necessario includere il parametro nei comandi. `--command-config client.properties`

Ad esempio, il comando seguente può essere utilizzato per modificare la proprietà di configurazione dell'argomento retention.bytes per impostare una conservazione illimitata:

```
<path-to-your-kafka-client-installation>/bin/kafka-configs.sh —bootstrap-
server <bootstrap_server_string> —command-config client.properties --entity-type topics
 --entity-name <topic_name> --alter --add-config retention.bytes=-1
```

In questo esempio, *`<bootstrap server string>`* sostituiscilo con l'endpoint del server bootstrap per il tuo cluster Amazon MSK Serverless e *`<topic_name>`* con il nome dell'argomento che desideri modificare.

Il `--command-config client.properties` parametro garantisce che lo strumento a riga di comando Kafka utilizzi le impostazioni di configurazione appropriate per comunicare con il cluster Amazon MSK Serverless.

Monitora i cluster MSK Serverless

Amazon MSK si integra con Amazon per CloudWatch consentirti di raccogliere, visualizzare e analizzare i parametri per il tuo cluster MSK Serverless. I parametri mostrati nella tabella seguente sono disponibili per tutti i cluster serverless. Poiché questi parametri sono pubblicati come punti di dati individuali per ogni partizione dell'argomento, consigliamo di visualizzarle come statistiche "SUM" per ottenere una visualizzazione a livello di argomento.

Amazon MSK pubblica i `PerSec` parametri con una frequenza di una volta CloudWatch al minuto. Ciò significa che la statistica "SUM" per un periodo di un minuto rappresenta accuratamente i dati al secondo per i parametri `PerSec`. Per raccogliere dati al secondo per un periodo superiore a un minuto, usa la seguente espressione matematica:. CloudWatch `m1 * 60/PERIOD(m1)`

Parametri disponibili al livello di monitoraggio DEFAULT

Nome	Quando visibile	Dimensioni	Descrizione
`BytesInPerSec`	Dopo che un produttore ha scritto su un argomento	Nome del cluster, argomento	Il numero di byte al secondo ricevuti dai client. Questo parametro è disponibile per ogni argomento.
`BytesOutPerSec`	Dopo che un gruppo di consumatori ha utilizzato un argomento	Nome del cluster, argomento	Il numero di byte al secondo inviati ai client. Questo parametro è disponibi le per ogni argomento.
`FetchMess ageConver sionsPerSec`	Dopo che un gruppo di consumatori ha utilizzato un argomento	Nome del cluster, argomento	Il numero di conversioni dei messaggi di recupero al secondo per l'argomen to.
`Estimated MaxTimeLag`	Dopo che un gruppo di consumatori ha utilizzato un argomento	Nome del cluster, gruppo di consumatori, argomento	Una stima temporale della metrica. MaxOffsetLag
`MaxOffsetLag`	Dopo che un gruppo di consumatori ha utilizzato un argomento	Nome del cluster, gruppo di consumatori, argomento	Il ritardo massimo di offset su tutte le partizioni di un argomento.

Nome	Quando visibile	Dimensioni	Descrizione
`MessagesInPerSec`	Dopo che un produttore ha scritto su un argomento	Nome del cluster, argomento	Il numero di messaggi in entrata al secondo per l'argomento.
`ProduceMessageConversionsPerSec`	Dopo che un produttore ha scritto su un argomento	Nome del cluster, argomento	Il numero di conversioni di messaggi di produzione al secondo per l'argomento.
`SumOffsetLag`	Dopo che un gruppo di consumatori ha utilizzato un argomento	Nome del cluster, gruppo di consumatori, argomento	Il ritardo di offset aggregato per tutte le partizioni di un argomento.

Visualizzazione dei parametri di MSK Serverless

1. Accedi a AWS Management Console e apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, in Parametri, scegli Tutti i parametri.
3. Nei parametri, cerca il termine **`kafka`**.
4. Per visualizzare parametri diversi, scegli AWS/Kafka / Nome del cluster, argomento oppure AWS/Kafka / Nome del cluster, gruppo di consumatori, argomento.

Comprendere MSK Connect

MSK Connect è una funzionalità di Amazon MSK che semplifica lo streaming di dati da e verso i cluster Apache Kafka a vantaggio degli sviluppatori. MSK Connect utilizza le versioni 2.7.1 o 3.7.x di Kafka Connect, che sono framework open source per connettere i cluster Apache Kafka con sistemi esterni come database, indici di ricerca e file system. Con MSK Connect, è possibile implementare connettori completamente gestiti creati per Kafka Connect che trasferiscono o estraggono dati da archivi di dati popolari come Amazon S3 e Amazon Service. OpenSearch È possibile implementare connettori sviluppati da terze parti come Debezium per eseguire lo streaming dei log delle modifiche dai database a un cluster Apache Kafka, oppure implementare un connettore esistente senza modifiche al codice. I connettori si dimensionano automaticamente in base alle variazioni di carico e ti verranno addebitate soltanto le risorse che effettivamente utilizzi.

Utilizza i connettori di origine per importare dati da sistemi esterni nei tuoi argomenti. Con i connettori sink, è possibile esportare i dati dai propri argomenti a sistemi esterni.

MSK Connect supporta connettori per qualsiasi cluster Apache Kafka con connettività a un Amazon VPC, che si tratti di un cluster MSK o di un cluster Apache Kafka ospitato in modo indipendente.

MSK Connect monitora continuamente l'integrità e lo stato di consegna dei connettori, corregge e gestisce l'hardware sottostante e dimensiona automaticamente i connettori in base alle variazioni della velocità di trasmissione effettiva.

Per le nozioni di base su MSK Connect, consulta la pagina the section called “Nozioni di base”.

Per ulteriori informazioni sulle AWS risorse che è possibile creare con MSK Connect, vedere the section called “Comprendi i connettori”the section called “Crea plugin personalizzati”, ethe section called “Conoscenza dei worker di MSK Connect”.

Per informazioni sull'API di MSK Connect, consulta la documentazione di riferimento sull'API di Amazon MSK Connect.

Vantaggi dell'utilizzo di Amazon MSK Connect

Apache Kafka è una delle piattaforme di streaming open source più utilizzate per l'acquisizione e l'elaborazione di flussi di dati in tempo reale. Con Apache Kafka, puoi disaccoppiare e scalare in modo indipendente le tue applicazioni che producono e consumano dati.

Kafka Connect è un componente importante per la creazione e l'esecuzione di applicazioni di streaming con Apache Kafka. Kafka Connect offre un modo standardizzato per lo spostamento dei dati tra Kafka e sistemi esterni. Kafka Connect è altamente scalabile e può gestire grandi volumi di dati Kafka Connect fornisce un potente set di operazioni e strumenti API per la configurazione, l'implementazione e il monitoraggio dei connettori che spostano i dati tra argomenti Kafka e sistemi esterni. Puoi utilizzare questi strumenti per personalizzare ed estendere le funzionalità di Kafka Connect per soddisfare le esigenze specifiche della tua applicazione di streaming.

Potresti incontrare difficoltà quando gestisci i cluster Apache Kafka Connect da soli o quando cerchi di migrare applicazioni open source Apache Kafka Connect verso. AWS Queste sfide includono il tempo necessario per configurare l'infrastruttura e implementare le applicazioni, gli ostacoli tecnici alla configurazione dei cluster Apache Kafka Connect autogestiti e il sovraccarico operativo amministrativo.

Per affrontare queste sfide, ti consigliamo di utilizzare Amazon Managed Streaming for Apache Kafka Connect (Amazon MSK Connect) per migrare le tue applicazioni open source Apache Kafka Connect verso. AWS Amazon MSK Connect semplifica l'utilizzo di Kafka Connect per lo streaming di dati da e verso cluster Apache Kafka e sistemi esterni, come database, indici di ricerca e file system.

Ecco alcuni dei vantaggi della migrazione ad Amazon MSK Connect:

- Eliminazione del sovraccarico operativo: Amazon MSK Connect elimina il carico operativo associato all'applicazione di patch, al provisioning e al ridimensionamento dei cluster Apache Kafka Connect. Amazon MSK Connect monitora continuamente lo stato dei cluster Connect e automatizza l'applicazione di patch e aggiornamenti di versione senza causare interruzioni ai carichi di lavoro.
- Riavvio automatico delle attività di Connect: Amazon MSK Connect può ripristinare automaticamente le attività non riuscite per ridurre le interruzioni della produzione. Gli errori delle attività possono essere causati da errori temporanei, come il superamento del limite di connessione TCP per Kafka e il ribilanciamento delle attività quando nuovi lavoratori si uniscono al gruppo di consumatori per i connettori sink.
- Scalabilità orizzontale e verticale automatica: Amazon MSK Connect consente all'applicazione del connettore di scalare automaticamente per supportare throughput più elevati. Amazon MSK Connect gestisce la scalabilità per te. È sufficiente specificare il numero di lavoratori nel gruppo di auto scaling e le soglie di utilizzo. Puoi utilizzare l'operazione dell'`UpdateConnector`API Amazon MSK Connect per scalare verticalmente verso l'alto o verso il basso la v CPUs tra 1 e 8 v CPUs per supportare un throughput variabile.

- Connettività di rete privata: Amazon MSK Connect si connette privatamente ai sistemi di origine e sink utilizzando nomi AWS PrivateLink DNS privati.

Guida introduttiva a MSK Connect

Questo è un step-by-step tutorial che utilizza AWS Management Console per creare un cluster MSK e un connettore sink che invia i dati dal cluster a un bucket S3.

Argomenti

Configurazione delle risorse necessarie per MSK Connect

In questo passaggio crei le seguenti risorse necessarie per questo scenario introduttivo:

- Un bucket Amazon S3 che funge da destinazione per la ricezione dei dati dal connettore.
- Un cluster MSK a cui inviare i dati. Il connettore leggerà i dati da questo cluster e li invierà al bucket S3 di destinazione.
- Una policy IAM che contiene le autorizzazioni per scrivere nel bucket S3 di destinazione.
- Un ruolo IAM che consente al connettore di scrivere nel bucket S3 di destinazione. A questo ruolo aggiungerai la policy IAM che crei.
- Un endpoint Amazon VPC per consentire l'invio di dati dall'Amazon VPC che include il cluster e il connettore ad Amazon S3.

Creazione del bucket S3

1. Accedi a AWS Management Console e apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
2. Seleziona Crea bucket.

3. Per il nome del bucket, specifica un nome descrittivo, ad esempio `amzn-s3-demo-bucket-mkc-tutorial`.
4. Scorri verso il basso e scegli Crea bucket.
5. Nell'elenco dei bucket, scegli il bucket appena creato.
6. Scegliere Create folder (Crea cartella).
7. Inserisci `tutorial` come nome della cartella, quindi scorri verso il basso e scegli Crea cartella.

Creazione del cluster

1. Aprire la console Amazon MSK a https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. Nel riquadro a sinistra, in Cluster MSK, scegli Cluster.
3. Scegli Create cluster (Crea cluster).
4. In Metodo di creazione, scegli Creazione personalizzata.
5. Come nome del cluster, specifica **`mkc-tutorial-cluster`**.
6. In Tipo di cluster, scegli Provisioned.
7. Scegli Next (Successivo).
8. In Rete, scegli un Amazon VPC. Seleziona quindi le zone di disponibilità e le sottoreti che desideri utilizzare. Ricorda il IDs VPC e le sottoreti Amazon che hai selezionato perché ne avrai bisogno più avanti in questo tutorial.
9. Scegli Next (Successivo).
10. In Metodi di controllo degli accessi, assicurati che sia selezionato soltanto Accesso non autenticato.
11. In Crittografia, assicurati che sia selezionato solo Non crittografato.
12. Continua con la procedura guidata, quindi scegli Crea cluster. In questo modo si accede alla pagina Dettagli per il cluster. In quella pagina, in Gruppi di sicurezza applicati, trova l'ID del gruppo di sicurezza. Ricorda quell'ID perché ne avrai bisogno più avanti in questo tutorial.

Per creare una policy IAM con autorizzazioni di scrittura nel bucket S3

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.

3. Scegliere Create Policy (Crea policy).
4. Nell'editor delle politiche, scegli JSON, quindi sostituisci il codice JSON nella finestra dell'editor con il seguente JSON.

   Nell'esempio seguente, sostituiscilo *<amzn-s3-demo-bucket-my-tutorial>* con il nome del tuo bucket S3.

   JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowListBucket",
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Resource": "arn:aws:s3:::<amzn-s3-demo-bucket-my-tutorial>"
    },
    {
      "Sid": "AllowObjectActions",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:AbortMultipartUpload",
        "s3:ListMultipartUploadParts",
        "s3:ListBucketMultipartUploads"
      ],
      "Resource": "arn:aws:s3:::<amzn-s3-demo-bucket-my-tutorial>/*"
    }
  ]
}
```

   Per istruzioni su come scrivere policy sicure, consulta. the section called “Controllo degli accessi IAM”
5. Scegli Next (Successivo).

6. Nella pagina Rivedi e crea, effettua le operazioni seguenti:

   a. Per Nome della politica, inserisci un nome descrittivo, ad esempio**`mkc-tutorial-policy`**.

   b. In Autorizzazioni definite in questa politica, rivedi e and/or modifica le autorizzazioni definite nella tua politica.

   c. (Facoltativo) Per facilitare l'identificazione, l'organizzazione o la ricerca della politica, scegli Aggiungi nuovo tag per aggiungere tag come coppie chiave-valore. Ad esempio, aggiungi un tag alla tua politica con la coppia chiave-valore e. **`Environment Test`**

      Per ulteriori informazioni sull'utilizzo dei tag, consulta Tags for AWS Identity and Access Management resources nella IAM User Guide.

7. Scegliere Create Policy (Crea policy).

Creazione del ruolo IAM che può scrivere nel bucket di destinazione

1. Nel pannello di navigazione della console IAM, scegli Ruoli, quindi scegli Crea ruolo.
2. Nella pagina Seleziona un'entità attendibile, esegui le operazioni seguenti:

   a. Per Trusted entity type (Tipo di entità attendibile), scegli Servizio AWS.

   b. Per Service o use case, scegli S3.

   c. In Caso d'uso, scegli S3.

3. Scegli Next (Successivo).
4. Nella pagina Add permissions (Aggiungi autorizzazioni), esegui le operazioni seguenti:

   a. Nella casella di ricerca sotto Politiche di autorizzazione, inserisci il nome della politica che hai creato in precedenza per questo tutorial. Ad esempio, mkc-tutorial-policy. Quindi, scegli la casella a sinistra del nome della politica.

   b. (Facoltativo) Impostare un limite delle autorizzazioni. Questa è una caratteristica avanzata disponibile per i ruoli di servizio, ma non per i ruoli collegati ai servizi. Per informazioni sull'impostazione di un limite di autorizzazioni, consulta Creating roles and attaching policies (console) nella IAM User Guide.

5. Scegli Next (Successivo).
6. Nella pagina Name, review, and create (Assegna un nome, rivedi e crea), esegui le operazioni seguenti:

   a. Per il nome del ruolo, inserisci un nome descrittivo, ad esempio. **`mkc-tutorial-role`**

> **Important**
>
> Quando assegni un nome a un ruolo, tieni presente quanto segue:
>
> - I nomi dei ruoli devono essere univoci all'interno del tuo Account AWS account e non possono essere resi unici per caso.
>
> Ad esempio, non creare ruoli denominati **`PRODROLE`** e **`prodrole`**. Quando il nome di un ruolo viene utilizzato in una policy o come parte di un ARN, il nome del ruolo fa distinzione tra maiuscole e minuscole, tuttavia quando un nome di ruolo viene visualizzato ai clienti nella console, ad esempio durante il processo di accesso, il nome del ruolo non fa distinzione tra maiuscole e minuscole.
> - Non è possibile modificare il nome del ruolo dopo averlo creato, in quanto altre entità possono fare riferimento al ruolo.

b. (Facoltativo) In Descrizione, inserisci una descrizione per il ruolo.

c. (Facoltativo) Per modificare i casi d'uso e le autorizzazioni per il ruolo, nel Passaggio 1: Seleziona le entità attendibili o nel Passaggio 2: Aggiungi le sezioni relative alle autorizzazioni, scegli Modifica.

d. (Facoltativo) Per facilitare l'identificazione, l'organizzazione o la ricerca del ruolo, scegli Aggiungi nuovo tag per aggiungere tag come coppie chiave-valore. Ad esempio, aggiungi un tag al tuo ruolo con la coppia chiave-valore e. **`ProductManager John`**

 Per ulteriori informazioni sull'utilizzo dei tag, consulta Tags for AWS Identity and Access Management resources nella IAM User Guide.

7. Verificare il ruolo e quindi scegliere Create role (Crea ruolo).

Autorizzazione di MSK Connect ad assumere il ruolo

1. Nella console IAM, nel riquadro a sinistra, in Gestione degli accessi, scegli Ruoli.
2. Trova il ruolo `mkc-tutorial-role` e selezionalo.
3. Nel Riepilogo del ruolo, scegli la scheda Relazioni di attendibilità.
4. Seleziona Modifica relazione di attendibilità.
5. Sostituisci la policy di attendibilità esistente con il JSON seguente.

JSON

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

6. Scegli Update Trust Policy (Aggiorna policy di trust).

Creazione di un endpoint VPC dal VPC del cluster ad Amazon S3

1. Apri la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Nel riquadro a sinistra, scegli Endpoint.
3. Seleziona Crea endpoint.
4. In Nome del servizio, scegli il servizio com.amazonaws.us-east-1.s3 e il tipo di Gateway.
5. Scegli il VPC del cluster, quindi seleziona la casella a sinistra della tabella di routing associata alle sottoreti del cluster.
6. Seleziona Crea endpoint.

Fase successiva

Crea un plugin personalizzato

Crea un plugin personalizzato

Un plug-in contiene il codice che definisce la logica del connettore. In questo passaggio crei un plug-in personalizzato con il codice per il connettore sink Amazon S3 Lenses. In un passaggio successivo, quando creerai il connettore MSK, specificherai che il relativo codice si trova in questo

plug-in personalizzato. È possibile utilizzare lo stesso plug-in per creare più connettori MSK con configurazioni diverse.

Creazione del plug-in personalizzato

1. Scarica il connettore S3.
2. Carica il file ZIP in un bucket S3 al quale puoi accedere. Per istruzioni su come caricare i file in Amazon S3, consulta la pagina Uploading objects nella Guida per l'utente di Amazon S3.
3. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/.
4. Nel riquadro a sinistra, espandi MSK Connect, quindi scegli Plug-in personalizzati.
5. Scegli Crea plug-in personalizzato.
6. Seleziona Sfoglia S3.
7. Nell'elenco dei bucket, trova il bucket in cui hai caricato il file ZIP e selezionalo.
8. Nell'elenco degli oggetti nel bucket, seleziona il pulsante di opzione a sinistra del file ZIP, quindi seleziona il pulsante con l'etichetta Scegli.
9. Inserisci `mkc-tutorial-plugin` come nome del plug-in personalizzato, quindi scegli Crea plug-in personalizzato.

Potrebbero essere necessari AWS alcuni minuti per completare la creazione del plug-in personalizzato. Al termine del processo di creazione, nella parte superiore della finestra del browser viene visualizzato il seguente messaggio in un banner.

```
Custom plugin mkc-tutorial-plugin was successfully created
The custom plugin was created. You can now create a connector using this custom plugin.
```

Fase successiva

Crea la macchina client e l'argomento Apache Kafka

Crea la macchina client e l'argomento Apache Kafka

In questo passaggio crei un' EC2 istanza Amazon da utilizzare come istanza client Apache Kafka. Quindi usi questa istanza per creare un argomento sul cluster.

Per creare un computer client

1. Apri la EC2 console Amazon all'indirizzo https://console.aws.amazon.com/ec2/.

2. Scegliere Launch Instances (Avvia istanze).
3. Inserisci un Nome per il computer client, ad esempio **`mkc-tutorial-client`**.
4. Lascia Amazon Linux 2 AMI (HVM) - Kernel 5.10, tipo di volume SSD selezionato per Tipo di Amazon Machine Image (AMI).
5. Scegli il tipo di istanza t2.xlarge.
6. In Coppia di chiavi (accesso), scegli Crea una nuova coppia di chiavi. Inserisci **`mkc-tutorial-key-pair`** in Nome della coppia di chiavi, quindi scegli Scarica coppia di chiavi. In alternativa, è possibile utilizzare una coppia di chiavi esistente.
7. Scegliere Launch Instance (Avvia istanza).
8. Scegliere View Instances (Vedi istanze). Quindi, nella colonna Gruppi di sicurezza, scegli il gruppo di sicurezza associato alla nuova istanza. Copia l'ID del gruppo di sicurezza e salvalo per un secondo momento.

Autorizzazione del client appena creato all'invio di dati al cluster

1. Apri la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Nel riquadro a sinistra, in Sicurezza, scegli Gruppi di sicurezza. Nella colonna ID del gruppo di sicurezza, trova il gruppo di sicurezza del cluster. Hai salvato l'ID di questo gruppo di sicurezza quando hai creato il cluster in the section called “Configurazione delle risorse necessarie per MSK Connect”. Scegli questo gruppo di sicurezza selezionando la casella a sinistra della riga. Assicurati che nessun altro gruppo di sicurezza sia selezionato contemporaneamente.
3. Nella sezione inferiore della pagina, scegli la scheda Regole in entrata.
4. Sceglere Edit inbound rules (Modifica regole in entrata).
5. In basso a sinistra dello schermo, scegli Aggiungi regola.
6. Nella nuova regola, scegliere All traffic (Tutto il traffico) nella colonna Type (Tipo) . Nel campo a destra della colonna Origine, inserisci l'ID del gruppo di sicurezza del computer client. Questo è l'ID del gruppo di sicurezza che hai salvato dopo aver creato il computer client.
7. Scegliere Salva regole. Il cluster MSK ora accetterà tutto il traffico proveniente dal client creato nella procedura precedente.

Per creare un argomento

1. Apri la EC2 console Amazon all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nella tabella delle istanze, scegli `mkc-tutorial-client`.

3. Nella parte superiore dello schermo, scegli Connetti, quindi segui le istruzioni per connetterti all'istanza.
4. Installa Java sull'istanza client eseguendo il seguente comando:

```
sudo yum install java-1.8.0
```

5. Eseguire il seguente comando per scaricare Apache Kafka.

```
wget https://archive.apache.org/dist/kafka/2.2.1/kafka_2.12-2.2.1.tgz
```

> **Note**
>
> Se desideri utilizzare un sito mirror diverso da quello utilizzato in questo comando, puoi sceglierne uno diverso sul sito Web di Apache .

6. Eseguire il comando seguente nella directory in cui è stato scaricato il file TAR nella fase precedente.

```
tar -xzf kafka_2.12-2.2.1.tgz
```

7. Passare alla directory kafka_2.12-2.2.1 .
8. Aprire la console Amazon MSK a https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
9. Nel riquadro a sinistra, scegli Cluster, quindi scegli il nome `mkc-tutorial-cluster`.
10. Scegli Visualizza le informazioni sul client.
11. Copia la stringa di connessione Non crittografato.
12. Seleziona Fatto.
13. Eseguite il comando seguente sull'istanza del client (`mkc-tutorial-client`), sostituendolo *`bootstrapServerString`* con il valore salvato quando avete visualizzato le informazioni sul client del cluster.

```
<path-to-your-kafka-installation>/bin/kafka-topics.sh --create --bootstrap-
server bootstrapServerString --replication-factor 2 --partitions 1 --topic mkc-
tutorial-topic
```

Se il comando va a buon fine, viene visualizzato il seguente messaggio: `Created topic mkc-tutorial-topic.`

Fase successiva

Crea connettore

Crea connettore

Questa procedura descrive come creare un connettore utilizzando AWS Management Console.

Creazione del connettore

1. Accedere a e aprire la console Amazon MSK da https://console.aws.amazon.com/msk/casa? AWS Management Console region=us-east-1#/home/.
2. Nel riquadro a sinistra, espandi MSK Connect, quindi scegli Connettori.
3. Scegli Create connector (Crea connettore).
4. Nell'elenco dei plugin, scegli `mkc-tutorial-plugin`, quindi scegli Avanti.
5. Per il nome del connettore, inserisci `mkc-tutorial-connector`.
6. Nell'elenco dei cluster, scegli `mkc-tutorial-cluster`.
7. Copia la configurazione seguente e incollala nel campo di configurazione del connettore.

   Assicurati di sostituire la regione con il codice del luogo in cui stai creando il connettore. Regione AWS Inoltre, sostituisci il nome del bucket Amazon S3 *<amzn-s3-demo-bucket-my-tutorial>* con il nome del tuo bucket nell'esempio seguente.

```
connector.class=io.confluent.connect.s3.S3SinkConnector
s3.region=us-east-1
format.class=io.confluent.connect.s3.format.json.JsonFormat
flush.size=1
schema.compatibility=NONE
tasks.max=2
topics=mkc-tutorial-topic
partitioner.class=io.confluent.connect.storage.partitioner.DefaultPartitioner
storage.class=io.confluent.connect.s3.storage.S3Storage
s3.bucket.name=<amzn-s3-demo-bucket-my-tutorial>
topics.dir=tutorial
```

8. In Autorizzazioni di accesso, scegli `mkc-tutorial-role`.
9. Scegli Next (Successivo). Nella pagina Sicurezza, scegli di nuovo Avanti.
10. Nella pagina Log, seleziona Avanti.

11. In Rivedi e crea, scegli Crea connettore.

Fase successiva

Invia dati al cluster MSK

Invia dati al cluster MSK

In questo passaggio si inviano i dati all'argomento Apache Kafka creato in precedenza, quindi si cercano gli stessi dati nel bucket S3 di destinazione.

Invio dei dati al cluster MSK

1. Nella cartella `bin` dell'installazione di Apache Kafka sull'istanza client, crea un file di testo denominato `client.properties` con i seguenti contenuti.

```
security.protocol=SASL_SSL
sasl.mechanism=AWS_MSK_IAM
```

2. Esegui il comando seguente per creare un produttore della console. Sostituisci *`BootstrapBrokerString`* con il valore ottenuto quando hai eseguito il comando precedente.

```
<path-to-your-kafka-installation>/bin/kafka-console-producer.sh --broker-
list BootstrapBrokerString --producer.config client.properties --topic mkc-
tutorial-topic
```

3. Immettere qualsiasi messaggio desiderato e premere Enter (Invio). Ripetere questa fase due o tre volte. Ogni volta che si immette una riga e si preme Enter (Invio), tale riga viene inviata al cluster Apache Kafka come un messaggio separato.
4. Cerca nel bucket Amazon S3 di destinazione per trovare i messaggi inviati nel passaggio precedente.

Comprendi i connettori

Un connettore integra sistemi esterni e servizi Amazon con Apache Kafka copiando continuamente i dati in streaming da un'origine dati nel cluster Apache Kafka o dal cluster in un sink di dati. Un connettore può anche eseguire operazioni logiche leggere come la trasformazione, la conversione del formato o il filtraggio dei dati prima di consegnarli a una destinazione. I connettori di origine

estraggono i dati da un'origine dati e li inviano al cluster, mentre i connettori sink estraggono i dati dal cluster e li inviano a un sink di dati.

Nel diagramma seguente viene illustrata l'architettura di un connettore. Un worker è un processo di macchina virtuale Java (JVM) che esegue la logica del connettore. Ogni worker crea una serie di attività che vengono eseguite in thread paralleli e si occupano di copiare i dati. Le attività non memorizzano lo stato e possono quindi essere avviate, interrotte o riavviate in qualsiasi momento per fornire una pipeline di dati resiliente e scalabile.

Comprendi la capacità dei connettori

La capacità totale di un connettore dipende dal numero di lavoratori del connettore e dal numero di MSK Connect Units (MCUs) per lavoratore. Ogni MCU rappresenta 1 vCPU di elaborazione e 4 GiB di memoria. La memoria MCU riguarda la memoria totale di un'istanza worker e non la memoria heap in uso.

Gli operatori di MSK Connect utilizzano gli indirizzi IP nelle sottoreti fornite dal cliente. Ogni lavoratore utilizza un indirizzo IP da una delle sottoreti fornite dal cliente. È necessario assicurarsi di disporre di un numero sufficiente di indirizzi IP disponibili nelle sottoreti fornite a una CreateConnector richiesta per tenere conto della capacità specificata, specialmente quando si scalano automaticamente i connettori in cui il numero di lavoratori può variare.

Per creare un connettore, è necessario scegliere tra una delle due modalità di capacità seguenti.

- Assegnato: scegli questa modalità se conosci i requisiti di capacità del tuo connettore. Specifica due valori:
  - Il numero di worker.
  - Il numero di dipendenti per lavoratore. MCUs
- Dimensionamento automatico: scegli questa modalità se i requisiti di capacità del connettore sono variabili o se non li conosci in anticipo. Quando utilizzi la modalità di scalabilità automatica, Amazon MSK Connect sostituisce la `tasks.max` proprietà del connettore con un valore proporzionale al numero di lavoratori in esecuzione nel connettore e al numero di lavoratori per lavoratore. MCUs

  Devi specificare tre set di valori:
  - Il numero minimo e massimo di worker.
  - Le percentuali di incremento e riduzione per l'utilizzo della CPU, determinate dal parametro `CpuUtilization`. Quando il parametro `CpuUtilization` del connettore supera la

percentuale di dimensionamento orizzontale, MSK Connect aumenta il numero di worker che utilizzano il connettore. Quando il parametro `CpuUtilization` scende al di sotto della percentuale di ridimensionamento, MSK Connect riduce il numero di worker. Il numero di worker rimane sempre compreso tra il numero minimo e massimo specificato al momento della creazione del connettore.

- MCUs Il numero di per lavoratore.

Per ulteriori informazioni sui worker, consulta la pagina the section called “Conoscenza dei worker di MSK Connect”. Per ulteriori informazioni sui parametri di MSK Connect, consulta la pagina the section called “Monitoraggio di MSK Connect”.

Creazione di un connettore

Questa procedura descrive come creare un connettore utilizzando AWS Management Console.

Creazione di un connettore utilizzando AWS Management Console

1. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/.
2. Nel riquadro a sinistra, in MSK Connect, scegli Connettori.
3. Scegli Create connector (Crea connettore).
4. Per creare il connettore, puoi scegliere se utilizzare un plug-in personalizzato esistente o creare innanzitutto un nuovo plug-in personalizzato. Per informazioni sui plug-in personalizzati e su come crearli, consulta la pagina the section called “Crea plugin personalizzati”. In questa procedura, supponiamo che tu abbia un plug-in personalizzato che desideri utilizzare. Nell'elenco dei plug-in personalizzati, trova quello che desideri utilizzare, seleziona la casella alla sua sinistra, quindi scegli Avanti.
5. Inserisci un nome e facoltativamente una descrizione.
6. Scegli il cluster a cui desideri connetterti.
7. Specifica la configurazione del connettore. I parametri di configurazione da specificare dipendono dal tipo di connettore che si desidera creare. Tuttavia, alcuni parametri sono comuni a tutti i connettori, ad esempio i parametri `connector.class` e `tasks.max`. Di seguito è riportato un esempio di configurazione per il connettore sink Amazon S3 Confluent.

```
connector.class=io.confluent.connect.s3.S3SinkConnector
tasks.max=2
topics=my-example-topic
s3.region=us-east-1
```

```
s3.bucket.name=amzn-s3-demo-bucket
flush.size=1
storage.class=io.confluent.connect.s3.storage.S3Storage
format.class=io.confluent.connect.s3.format.json.JsonFormat
partitioner.class=io.confluent.connect.storage.partitioner.DefaultPartitioner
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.kafka.connect.storage.StringConverter
schema.compatibility=NONE
```

8. Successivamente, configura la capacità del connettore. È possibile scegliere tra due modalità di capacità: assegnata e con dimensionamento automatico. Per informazioni su queste due opzioni, consulta the section called “Comprendi la capacità dei connettori”.
9. Scegli la configurazione del worker predefinita o una configurazione del worker personalizzata. Per informazioni sulla creazione di configurazioni del worker personalizzate, consulta la pagina the section called “Conoscenza dei worker di MSK Connect”.
10. Successivamente, specifica il ruolo di esecuzione del servizio. Questo deve essere un ruolo IAM che MSK Connect può assumere e che concede al connettore tutte le autorizzazioni necessarie per accedere alle risorse necessarie. AWS Tali autorizzazioni dipendono dalla logica del connettore. Per informazioni su come creare questo ruolo, consulta the section called “Comprendi il ruolo di esecuzione del servizio”.
11. Scegli Avanti, esamina le informazioni di sicurezza, quindi scegli nuovamente Avanti.
12. Specifica le opzioni di registrazione desiderate, quindi scegli Avanti. Per ulteriori informazioni sulla registrazione, consulta the section called “Registrazione”.
13. Scegli Create connector (Crea connettore).

Per utilizzare l'API MSK Connect per creare un connettore, vedere CreateConnector.

È possibile utilizzare l'`UpdateConnector`API per modificare la configurazione del connettore. Per ulteriori informazioni, consulta the section called “Aggiorna un connettore”.

Aggiorna un connettore

Questa procedura descrive come aggiornare la configurazione di un connettore MSK Connect esistente utilizzando. AWS Management Console

Aggiornamento della configurazione del connettore utilizzando il AWS Management Console

1. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/.

2. Nel riquadro a sinistra, in MSK Connect, scegli Connettori.
3. Seleziona un connettore esistente.
4. Scegli Modifica configurazione del connettore.
5. Aggiorna la configurazione del connettore. Non è possibile sovrascrivere `connector.class` l'utilizzo UpdateConnector di. L'esempio seguente mostra una configurazione di esempio per il connettore Confluent Amazon S3 Sink.

```
connector.class=io.confluent.connect.s3.S3SinkConnector
tasks.max=2
topics=my-example-topic
s3.region=us-east-1
s3.bucket.name=amzn-s3-demo-bucket
flush.size=1
storage.class=io.confluent.connect.s3.storage.S3Storage
format.class=io.confluent.connect.s3.format.json.JsonFormat
partitioner.class=io.confluent.connect.storage.partitioner.DefaultPartitioner
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.kafka.connect.storage.StringConverter
schema.compatibility=NONE
```

6. Scegli Invia.
7. È quindi possibile monitorare lo stato corrente dell'operazione nella scheda Operazioni del connettore.

Per utilizzare l'API MSK Connect per aggiornare la configurazione di un connettore, vedere UpdateConnector.

Connessione dai connettori

Le seguenti best practice possono migliorare le prestazioni della connettività ad Amazon MSK Connect.

Non si sovrappongono IPs per Amazon VPC, peering o Transit Gateway

Se utilizzi il peering Amazon VPC o Transit Gateway con Amazon MSK Connect, non configurare il connettore per raggiungere le risorse VPC peerizzate con gli intervalli CIDR: IPs

- "10.99.0.0/16"
- "192.168.0.0/16"

- "172.21.0.0/16"

Crea plugin personalizzati

Un plugin è una AWS risorsa che contiene il codice che definisce la logica del connettore. Quando si crea il plug-in, si carica un file JAR (o un file ZIP che contiene uno o più file JAR) in un bucket S3 e si specifica la posizione del bucket. Quando si crea un connettore, si specifica il plug-in che si desidera che MSK Connect utilizzi. La relazione tra i plugin e i connettori è one-to-many: È possibile creare uno o più connettori dallo stesso plugin.

Per informazioni su come sviluppare il codice per un connettore, consulta la pagina Connector Development Guide nella documentazione di Apache Kafka.

Creazione di un plug-in personalizzato utilizzando il AWS Management Console

1. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/.
2. Nel riquadro a sinistra, in MSK Connect, scegli Plug-in personalizzati.
3. Scegli Crea plug-in personalizzato.
4. Seleziona Sfoglia S3.
5. Nell'elenco dei bucket S3, scegli il bucket contenente il file JAR o ZIP per il plug-in.
6. Nell'elenco degli oggetti, seleziona la casella a sinistra del file JAR o ZIP per il plug-in, quindi seleziona Scegli.
7. Scegli Crea plug-in personalizzato.

Per utilizzare l'API MSK Connect per creare un plug-in personalizzato, vedere CreateCustomPlugin.

Conoscenza dei worker di MSK Connect

Un worker è un processo di macchina virtuale Java (JVM) che esegue la logica del connettore. Ogni worker crea una serie di attività che vengono eseguite in thread paralleli e si occupano di copiare i dati. Le attività non memorizzano lo stato e possono quindi essere avviate, interrotte o riavviate in qualsiasi momento per fornire una pipeline di dati resiliente e scalabile. Le modifiche al numero di worker, dovute a un evento di dimensionamento o a guasti imprevisti, vengono rilevate automaticamente dai worker rimanenti. Essi si coordinano per riequilibrare le attività tra il gruppo di worker rimanenti. I worker di Connect utilizzano i gruppi di consumatori di Apache Kafka per tali operazioni di coordinamento e riequilibrio.

Se i requisiti di capacità del connettore sono variabili o difficili da stimare, è possibile consentire a MSK Connect di dimensionare il numero di worker in base alle esigenze entro un limite inferiore e un limite superiore specificati. In alternativa, è possibile specificare il numero esatto di worker da utilizzare per l'esecuzione della logica di connessione. Per ulteriori informazioni, consulta the section called “Comprendi la capacità dei connettori”.

Gli operatori di MSK Connect utilizzano gli indirizzi IP

Gli operatori di MSK Connect utilizzano gli indirizzi IP nelle sottoreti fornite dal cliente. Ogni lavoratore utilizza un indirizzo IP da una delle sottoreti fornite dal cliente. È necessario assicurarsi di disporre di un numero sufficiente di indirizzi IP disponibili nelle sottoreti fornite a una CreateConnector richiesta per tenere conto della capacità specificata, specialmente quando si scalano automaticamente i connettori in cui il numero di lavoratori può variare.

Configurazione dei worker predefinita

MSK Connect fornisce la seguente configurazione predefinita per i worker:

```
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.kafka.connect.storage.StringConverter
```

Proprietà di configurazione dei worker supportate

MSK Connect fornisce una configurazione predefinita per i worker. Se lo desideri, puoi creare una configurazione dei worker personalizzata da utilizzare con i connettori. L'elenco seguente include informazioni sulle proprietà di configurazione dei worker supportate o meno da Amazon MSK Connect.

- Sono obbligatorie solo le proprietà `key.converter` e `value.converter`.
- MSK Connect supporta le seguenti proprietà di configurazione di `producer.`.

```
producer.acks
producer.batch.size
producer.buffer.memory
producer.compression.type
producer.enable.idempotence
producer.key.serializer
producer.linger.ms
producer.max.request.size
producer.metadata.max.age.ms
```

```
producer.metadata.max.idle.ms
producer.partitioner.class
producer.reconnect.backoff.max.ms
producer.reconnect.backoff.ms
producer.request.timeout.ms
producer.retry.backoff.ms
producer.value.serializer
```

- MSK Connect supporta le seguenti proprietà di configurazione di `consumer.`.

```
consumer.allow.auto.create.topics
consumer.auto.offset.reset
consumer.check.crcs
consumer.fetch.max.bytes
consumer.fetch.max.wait.ms
consumer.fetch.min.bytes
consumer.heartbeat.interval.ms
consumer.key.deserializer
consumer.max.partition.fetch.bytes
consumer.max.poll.interval.ms
consumer.max.poll.records
consumer.metadata.max.age.ms
consumer.partition.assignment.strategy
consumer.reconnect.backoff.max.ms
consumer.reconnect.backoff.ms
consumer.request.timeout.ms
consumer.retry.backoff.ms
consumer.session.timeout.ms
consumer.value.deserializer
```

- Sono supportate tutte le altre proprietà di configurazione che non iniziano con i prefissi `producer.` o `consumer.`, ad eccezione delle seguenti proprietà.

```
access.control.
admin.
admin.listeners.https.
client.
connect.
inter.worker.
internal.
listeners.https.
metrics.
metrics.context.
```

```
rest.
sasl.
security.
socket.
ssl.
topic.tracking.
worker.
bootstrap.servers
config.storage.topic
connections.max.idle.ms
connector.client.config.override.policy
group.id
listeners
metric.reporters
plugin.path
receive.buffer.bytes
response.http.headers.config
scheduled.rebalance.max.delay.ms
send.buffer.bytes
status.storage.topic
```

Per ulteriori informazioni sulle proprietà di configurazione dei worker e su cosa rappresentano, consulta la pagina Kafka Connect Configs nella documentazione di Apache Kafka.

Creazione di una configurazione dei worker personalizzata

Questa procedura illustra come creare una configurazione dei worker personalizzata tramite la AWS Management Console.

Creazione di una configurazione dei worker personalizzata tramite la AWS Management Console

1. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/.
2. Nel riquadro a sinistra, in MSK Connect, scegli Configurazioni del worker.
3. Seleziona Configurazione del worker.
4. Inserisci un nome e una descrizione opzionale, quindi aggiungi le proprietà e i valori su cui desideri impostarli.
5. Seleziona Configurazione del worker.

Per creare una configurazione del worker tramite l'API di MSK Connect, consulta CreateWorkerConfigurationla sezione.

Gestisci gli offset dei connettori di origine utilizzando `offset.storage.topic`

Questa sezione fornisce informazioni per aiutarti a gestire gli offset dei connettori di origine tramite l'argomento di archiviazione degli offset. L'argomento di archiviazione degli offset è un argomento interno che Kafka Connect utilizza per archiviare gli offset di configurazione dei connettori e delle attività.

Considerazioni

Durante la gestione degli offset del connettore di origine, tieni in considerazione i seguenti aspetti.

- Per specificare un argomento archiviazione degli offset, fornisci il nome dell'argomento Kafka in cui gli offset dei connettori vengono archiviati come valore di `offset.storage.topic` nella configurazione del worker.
- Presta attenzione quando apporti modifiche alla configurazione di un connettore. Se un connettore di origine utilizza i valori della configurazione per record di offset chiave, la modifica dei valori di configurazione può causare un comportamento indesiderato del connettore. Ti consigliamo di fare riferimento alla documentazione del tuo plug-in per informazioni.
- Personalizza il numero predefinito di partizioni: oltre a personalizzare la configurazione del worker mediante l'aggiunta di `offset.storage.topic`, è possibile personalizzare il numero di partizioni per gli argomenti di archiviazione degli offset e degli stati. Le partizioni predefinite per gli argomenti interni sono le seguenti.
  - `config.storage.topic`: 1, non configurabile, deve essere un argomento a partizione singola
  - `offset.storage.topic`: 25, configurabile fornendo `offset.storage.partitions`
  - `status.storage.topic`: 5, configurabile fornendo `status.storage.partitions`
- Eliminazione manuale degli argomenti: Amazon MSK Connect crea nuovi argomenti interni di Kafka Connect (il nome dell'argomento inizia con `__amazon_msk_connect`) a ogni implementazione di connettori. I vecchi argomenti associati ai connettori eliminati non vengono rimossi automaticamente perché gli argomenti interni, ad esempio `offset.storage.topic`, possono essere riutilizzati tra i connettori. Tuttavia, è possibile eliminare manualmente gli argomenti interni non utilizzati creati da MSK Connect. Gli argomenti interni sono denominati secondo il formato `__amazon_msk_connect_<offsets|status|configs>_`*`connector_name_connector_id`*.

Per eliminare gli argomenti interni, è possibile utilizzare l'espressione regolare `__amazon_msk_connect_<offsets|status|configs>_connector_name_connector_id`. Evita di eliminare un argomento interno attualmente utilizzato da un connettore in esecuzione.

- Utilizzo dello stesso nome per gli argomenti interni creati d MSK Connect: se desideri riutilizzare l'argomento di archiviazione degli offset per utilizzare gli offset di un connettore creato in precedenza, dovrai assegnare al nuovo connettore lo stesso nome di quello precedente. Nella configurazione del worker, è possibile impostare la proprietà `offset.storage.topic` per assegnare lo stesso nome a `offset.storage.topic` e riutilizzarlo tra connettori diversi. Questa configurazione è descritta nella sezione Gestione degli offset dei connettori. MSK Connect non consente a connettori diversi di condividere `config.storage.topic` e `status.storage.topic`. Questi argomenti vengono creati ogni volta che si crea un nuovo connettore in MSKC. Vengono denominati automaticamente secondo il formato `__amazon_msk_connect_<status|configs>_connector_name_connector_id` e quindi sono diversi per ciascuno dei connettori creati.

Utilizzo dell'argomento predefinito per l'archiviazione degli offset

Per impostazione predefinita, Amazon MSK Connect genera un nuovo argomento di archiviazione degli offset sul cluster Kafka per ogni connettore creato. MSK costruisce il nome dell'argomento predefinito utilizzando parti dell'ARN del connettore. Ad esempio, `__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-4abc-8be8-c657f7e4ff32-2`.

Argomento personalizzato per l'archiviazione degli offset

Per garantire la continuità degli offset tra i connettori di origine, puoi utilizzare un argomento di archiviazione degli offset a tua scelta anziché l'argomento predefinito. La definizione di un argomento di archiviazione degli offset consente di eseguire attività come la creazione di un connettore di origine che riprenda la lettura dall'ultimo offset di un connettore precedente.

Per definire un argomento di archiviazione degli offset, è necessario fornire un valore per la proprietà `offset.storage.topic` nella configurazione del worker prima di creare un connettore. Se si desidera riutilizzare l'argomento di archiviazione degli offset per utilizzare gli offset di un connettore creato in precedenza, è necessario assegnare al nuovo connettore lo stesso nome di quello precedente. Se si crea un argomento di archiviazione degli offset personalizzato, è necessario impostare `cleanup.policy` su `compact` nella configurazione dell'argomento.

> **Note**
>
> Se si specifica un argomento di archiviazione degli offset quando si crea un connettore sink, MSK Connect crea l'argomento, se non esiste ancora. Tuttavia, l'argomento non verrà utilizzato per archiviare gli offset dei connettori.
> Gli offset dei connettori sink vengono invece gestiti utilizzando il protocollo del gruppo di consumatori di Kafka. Ogni connettore sink crea un gruppo denominato `connect-{CONNECTOR_NAME}`. Finché esiste il gruppo di consumatori, tutti i connettori sink creati successivamente con lo stesso valore di `CONNECTOR_NAME` continueranno dall'ultimo offset confermato.

Example : definizione di un argomento di archiviazione degli offset per ricreare un connettore di origine con una configurazione aggiornata

Supponi di avere un connettore Change Data Capture (CDC) e di voler modificare la configurazione del connettore senza perdere il posto nel flusso CDC. Non è possibile aggiornare la configurazione esistente del connettore, ma è possibile eliminare il connettore e crearne uno nuovo con lo stesso nome. Per indicare al nuovo connettore da dove iniziare a leggere il flusso CDC, puoi specificare l'argomento di archiviazione degli offset del vecchio connettore nella configurazione del worker. Di seguito viene illustrato come realizzare tale operazione.

1. Sul computer client, esegui il comando seguente per trovare il nome dell'argomento archiviazione degli offset del connettore. Sostituisci *<bootstrapBrokerString>* con la stringa del broker di bootstrap del cluster. Per istruzioni su come recuperare la stringa del broker di bootstrap, consulta la pagina Ottieni i broker bootstrap per un cluster Amazon MSK.

   ```
   <path-to-your-kafka-installation>/bin/kafka-topics.sh --list --bootstrap-
   server <bootstrapBrokerString>
   ```

   L'output seguente mostra un elenco di tutti gli argomenti del cluster, inclusi gli argomenti predefiniti relativi ai connettori interni. In questo esempio, il connettore CDC esistente utilizza l'argomento di archiviazione degli offset predefinito creato da MSK Connect. Questo è il motivo per cui l'argomento di archiviazione degli offset è chiamato `__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-4abc-8be8-c657f7e4ff32-2`.

   ```
   __consumer_offsets
   ```

```
__amazon_msk_canary
__amazon_msk_connect_configs_my-mskc-connector_12345678-09e7-4abc-8be8-
c657f7e4ff32-2
__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-4abc-8be8-
c657f7e4ff32-2
__amazon_msk_connect_status_my-mskc-connector_12345678-09e7-4abc-8be8-
c657f7e4ff32-2
my-msk-topic-1
my-msk-topic-2
```

2. Apri la console Amazon MSK all'indirizzo https://console.aws.amazon.com/msk/.
3. Scegli il connettore dall'elenco Connettori. Copia e salva il contenuto del campo Configurazione del connettore in modo da poterlo modificare e utilizzare per creare il nuovo connettore.
4. Scegli Elimina per confermare l'eliminazione. Inserisci il nome del connettore nel campo di immissione del testo per confermare l'eliminazione.
5. Crea una configurazione di worker personalizzata con valori adatti al tuo scenario. Per istruzioni, consultare Creazione di una configurazione dei worker personalizzata.

   Nella configurazione del worker, è necessario specificare il nome dell'argomento di archiviazione degli offset recuperato in precedenza come valore di `offset.storage.topic`, come nella configurazione seguente.

   ```
   config.providers.secretManager.param.aws.region=eu-west-3
   key.converter=<org.apache.kafka.connect.storage.StringConverter>
   value.converter=<org.apache.kafka.connect.storage.StringConverter>
   config.providers.secretManager.class=com.github.jcustenborder.kafka.config.aws.SecretsManag
   config.providers=secretManager
   offset.storage.topic=__amazon_msk_connect_offsets_my-mskc-
   connector_12345678-09e7-4abc-8be8-c657f7e4ff32-2
   ```

6. **Important**

   Al nuovo connettore deve essere assegnato lo stesso nome del vecchio connettore.

   Crea un nuovo connettore utilizzando la configurazione del worker impostata nel passaggio precedente. Per istruzioni, consulta Creazione di un connettore.

Tutorial: esternalizzazione di informazioni sensibili utilizzando provider di configurazione

Questo esempio mostra come esternalizzare le informazioni sensibili per Amazon MSK Connect utilizzando un provider di configurazione open source. Un provider di configurazione consente di specificare variabili anziché testo non crittografato in una configurazione di connettore o di worker e i worker in esecuzione nel connettore risolvono queste variabili in fase di runtime. Ciò impedisce che le credenziali e altri segreti vengano archiviati in testo non crittografato. Il provider di configurazione nell'esempio supporta il recupero dei parametri di configurazione da AWS Secrets Manager, Amazon S3 e Systems Manager (SSM). Nel passaggio 2, viene illustrato come configurare l'archiviazione e il recupero di informazioni sensibili per il servizio che desideri configurare.

Considerazioni

Considera quanto segue durante l'utilizzo del provider di configurazione MSK con Amazon MSK Connect:

- Quando utilizzi i provider di configurazione, assegna le autorizzazioni appropriate al ruolo di esecuzione del servizio IAM.
- Definisci i provider di configurazione nelle configurazioni dei worker e la rispettiva implementazione nella configurazione del connettore.
- Se un plug-in non definisce i valori di configurazione sensibili come segreti, tali valori possono apparire nei log dei connettori. Kafka Connect tratta i valori di configurazione non definiti allo stesso modo di qualsiasi altro valore non crittografato. Per ulteriori informazioni, consulta Impedire la visualizzazione di segreti nei log dei connettori.
- Per impostazione predefinita, spesso MSK Connect riavvia un connettore se questo utilizza un provider di configurazione. Per disattivare questo comportamento di riavvio, è possibile impostare il valore di `config.action.reload` su `none` nella configurazione del connettore.

Crea un plug-in personalizzato e caricalo su S3

Per creare un plugin personalizzato, crea un file zip che contenga il connettore ed esegui i seguenti comandi sul tuo computer locale. msk-config-provider

Creazione di un plug-in personalizzato utilizzando una finestra di terminale e Debezium come connettore

Usa la AWS CLI per eseguire comandi come superutente con credenziali che ti consentono di accedere al tuo bucket S3. AWS Per informazioni sull'installazione e la configurazione della AWS CLI, consulta Guida introduttiva alla AWS CLI nella Guida per l'utente.AWS Command Line Interface Per informazioni sull'uso della AWS CLI con Amazon S3, consulta Using Amazon S3 with the AWS CLI nella Guida per l'utente.AWS Command Line Interface

1. In una finestra del terminale, crea una cartella denominata `custom-plugin` nel tuo spazio di lavoro tramite il seguente comando.

   ```
   mkdir custom-plugin && cd custom-plugin
   ```

2. Scarica l'ultima versione stabile del plug-in per il connettore MySQL dal sito di Debezium tramite il seguente comando.

   ```
   wget https://repo1.maven.org/maven2/io/debezium/debezium-connectormysql/
   2.2.0.Final/debezium-connector-mysql-2.2.0.Final-plugin.tar.gz
   ```

   Estrai il file gzip scaricato nella cartella `custom-plugin` tramite il seguente comando.

   ```
   tar xzf debezium-connector-mysql-2.2.0.Final-plugin.tar.gz
   ```

3. Scarica il file zip del provider di configurazione MSK tramite il seguente comando.

   ```
   wget https://github.com/aws-samples/msk-config-providers/releases/download/r0.4.0/
   msk-config-providers-0.4.0-with-dependencies.zip
   ```

   Estrai il file zip scaricato nella cartella `custom-plugin` tramite il seguente comando.

   ```
   unzip msk-config-providers-0.4.0-with-dependencies.zip
   ```

4. Comprimi il contenuto del provider di configurazione MSK del passaggio precedente e del connettore personalizzato in un unico file denominato `custom-plugin.zip`.

   ```
   zip -r ../custom-plugin.zip *
   ```

5. Carica il file su S3 per utilizzarlo come riferimento in seguito.

```
aws s3 cp ../custom-plugin.zip s3:<S3_URI_BUCKET_LOCATION>
```

6. Sulla console Amazon MSK, nella sezione MSK Connect, scegli Custom Plugin, quindi scegli Crea plug-in personalizzato e sfoglia il bucket s3: **< *S3_URI_BUCKET_LOCATION* >** S3 per selezionare il file ZIP del plug-in personalizzato che hai appena caricato.

7. Inserisci **`debezium-custom-plugin`** come nome del plug-in. Facoltativamente, inserisci una descrizione e scegli Crea plug-in personalizzato.

Configura parametri e autorizzazioni per diversi provider

È possibile configurare i valori dei parametri in questi tre servizi:

- Secrets Manager
- Systems Manager Parameter Store
- S3 - Simple Storage Service

Scegli una delle schede seguenti per istruzioni sulla configurazione dei parametri e delle autorizzazioni pertinenti per quel servizio.

Configure in Secrets Manager

Configurazione dei valori dei parametri in Secrets Manager

1. Apri la console Secrets Manager.
2. Crea un nuovo segreto per archiviare le credenziali o i segreti. Per istruzioni, consulta Creare un AWS Secrets Manager segreto nella Guida per l'AWS Secrets Manager utente.
3. Copia l'ARN del segreto.
4. Aggiungi le autorizzazioni di Secrets Manager dalla seguente policy di esempio al tuo ruolo di esecuzione del servizio. Sostituisci l'esempio ARN,`arn:aws:secretsmanager:us-east-1:123456789012:secret:MySecret-1234`, con l'ARN del tuo segreto.
5. Aggiungi le istruzioni per la configurazione del worker e il connettore.

```
{
```

```
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "secretsmanager:GetResourcePolicy",
                    "secretsmanager:GetSecretValue",
                    "secretsmanager:DescribeSecret",
                    "secretsmanager:ListSecretVersionIds"
                ],
                "Resource": [
                "arn:aws:secretsmanager:us-
east-1:123456789012:secret:MySecret-1234"
                ]
            }
        ]
    }
```

6. Per utilizzare il provider di configurazione Secrets Manager, copia le seguenti righe di codice nella casella di testo della configurazione del worker nel passaggio 3:

```
# define name of config provider:

config.providers = secretsmanager

# provide implementation classes for secrets manager:

config.providers.secretsmanager.class =
 com.amazonaws.kafka.config.providers.SecretsManagerConfigProvider

# configure a config provider (if it needs additional initialization), for
 example you can provide a region where the secrets or parameters are located:

config.providers.secretsmanager.param.region = us-east-1
```

7. Per il provider di configurazione Secrets Manager, copia le seguenti righe di codice nella configurazione del connettore nel passaggio 4.

```
#Example implementation for secrets manager variable
database.user=${secretsmanager:MSKAuroraDBCredentials:username}
database.password=${secretsmanager:MSKAuroraDBCredentials:password}
```

È possibile utilizzare il passaggio precedente anche con altri provider di configurazione.

Configure in Systems Manager Parameter Store

Configurazione dei valori dei parametri in Archivio dei parametri Systems Manager

1. Aprire la console Systems Manager.
2. Nel riquadro di navigazione, selezionare Parameter Store (Archivio parametri).
3. Crea un nuovo parametro da archiviare in Systems Manager. Per istruzioni, vedere Create a Systems Manager (console) nella Guida per l' AWS Systems Manager utente.
4. Copia l'ARN del parametro.
5. Aggiungi le autorizzazioni di Systems Manager dalla seguente policy di esempio al tuo ruolo di esecuzione del servizio. Sostituisci *<arn:aws:ssm:us-east-1:123456789000:parameter/MyParameterName>* con l'ARN del tuo parametro.

```
{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "VisualEditor0",
                "Effect": "Allow",
                "Action": [
                    "ssm:GetParameterHistory",
                    "ssm:GetParametersByPath",
                    "ssm:GetParameters",
                    "ssm:GetParameter"
                ],
                "Resource": "arn:aws:ssm:us-east-1:123456789000:parameter/
MyParameterName"
            }
        ]
    }
```

6. Per utilizzare il provider di configurazione Archivio dei parametri, copia le seguenti righe di codice nella casella di testo della configurazione del worker nel passaggio 3:

```
# define name of config provider:

config.providers = ssm
```

```
# provide implementation classes for parameter store:

config.providers.ssm.class =
 com.amazonaws.kafka.config.providers.SsmParamStoreConfigProvider

# configure a config provider (if it needs additional initialization), for
 example you can provide a region where the secrets or parameters are located:

config.providers.ssm.param.region = us-east-1
```

7. Per il provider di configurazione Archivio dei parametri, copia le seguenti righe di codice nella configurazione del connettore nel passaggio 5.

```
#Example implementation for parameter store variable
schema.history.internal.kafka.bootstrap.servers=
${ssm::MSKBootstrapServerAddress}
```

È possibile utilizzare i due passaggi precedenti anche con altri provider di configurazione.

Configure in Amazon S3

Per configurare objects/files in Amazon S3

1. Apri la console Amazon S3.
2. Carica l'oggetto in un bucket S3. Per istruzioni, consulta la pagina Uploading objects.
3. Copia l'ARN dell'oggetto.
4. Aggiungi le autorizzazioni di Amazon S3 Object Read dalla seguente policy di esempio al tuo ruolo di esecuzione del servizio. Sostituisci l'esempio ARN,`arn:aws:s3:::<MY_S3_BUCKET/path/to/custom-plugin.zip>`, con l'ARN del tuo oggetto.

```
{
    "Version": "2012-10-17",
    "Statement": [
            {
                "Sid": "VisualEditor0",
                "Effect": "Allow",
                "Action": "s3:GetObject",
```

```
                "Resource": "arn:aws:s3:::<MY_S3_BUCKET/path/to/custom-
plugin.zip>"
            }
        ]
    }
```

5. Per utilizzare il provider di configurazione Amazon S3, copia le seguenti righe di codice nella casella di testo della configurazione del worker nel passaggio 3:

```
# define name of config provider:

config.providers = s3import
# provide implementation classes for S3:

config.providers.s3import.class =
 com.amazonaws.kafka.config.providers.S3ImportConfigProvider
```

6. Per il provider di configurazione Amazon S3, copia le seguenti righe di codice nella configurazione del connettore nel passaggio 4.

```
#Example implementation for S3 object

database.ssl.truststore.location = ${s3import:us-west-2:my_cert_bucket/path/to/
trustore_unique_filename.jks}
```

È possibile utilizzare i due passaggi precedenti anche con altri provider di configurazione.

Creazione di una configurazione del worker personalizzata con informazioni sul proprio provider di configurazione

1. Seleziona Configurazioni dei worker nella sezione Amazon MSK Connect.
2. Seleziona Crea configurazione del worker.
3. Inserisci `SourceDebeziumCustomConfig` nella casella di testo Nome della configurazione del worker. La descrizione è facoltativa.
4. Copia il codice di configurazione pertinente in base ai provider desiderati e incollalo nella casella di testo Configurazione del worker.
5. Questo è un esempio di configurazione del worker per tutti e tre i provider:

```
key.converter=org.apache.kafka.connect.storage.StringConverter
key.converter.schemas.enable=false
value.converter=org.apache.kafka.connect.json.JsonConverter
value.converter.schemas.enable=false
offset.storage.topic=offsets_my_debezium_source_connector

# define names of config providers:

config.providers=secretsmanager,ssm,s3import

# provide implementation classes for each provider:

config.providers.secretsmanager.class    =
 com.amazonaws.kafka.config.providers.SecretsManagerConfigProvider
config.providers.ssm.class               =
 com.amazonaws.kafka.config.providers.SsmParamStoreConfigProvider
config.providers.s3import.class          =
 com.amazonaws.kafka.config.providers.S3ImportConfigProvider

# configure a config provider (if it needs additional initialization), for example
 you can provide a region where the secrets or parameters are located:


config.providers.secretsmanager.param.region = us-east-1
config.providers.ssm.param.region = us-east-1
```

6. Fai clic su Crea configurazione del worker.

Crea il connettore

1. Crea un nuovo connettore seguendo le istruzioni riportate nella sezione Creazione di un nuovo connettore.
2. Scegli il file `custom-plugin.zip` che hai caricato nel tuo bucket S3 in ??? come origine del plug-in personalizzato.
3. Copia il codice di configurazione pertinente in base ai provider desiderati e incollalo nel campo Configurazione del cluster.
4. Questo è un esempio della configurazione dei connettori per tutti e tre i provider:

```
#Example implementation for parameter store variable
```

```
schema.history.internal.kafka.bootstrap.servers=${ssm::MSKBootstrapServerAddress}

#Example implementation for secrets manager variable
database.user=${secretsmanager:MSKAuroraDBCredentials:username}
database.password=${secretsmanager:MSKAuroraDBCredentials:password}

#Example implementation for Amazon S3 file/object
database.ssl.truststore.location = ${s3import:us-west-2:my_cert_bucket/path/to/
trustore_unique_filename.jks}
```

5. Seleziona Usa una configurazione personalizzata e scegli SourceDebeziumCustomConfigdal menu a discesa Worker Configuration.
6. Segui i passaggi rimanenti indicati nelle istruzioni nella sezione Creazione di un connettore.

Ruoli e policy IAM per MSK Connect

Questa sezione ti aiuta a configurare le politiche e i ruoli IAM appropriati per distribuire e gestire in modo sicuro Amazon MSK Connect all'interno AWS del tuo ambiente. Le sezioni seguenti spiegano il ruolo di esecuzione del servizio che deve essere utilizzato con MSK Connect, inclusa la politica di attendibilità richiesta e le autorizzazioni aggiuntive necessarie per la connessione a un cluster MSK autenticato tramite IAM. La pagina fornisce anche esempi di policy IAM complete per garantire l'accesso completo alle funzionalità di MSK Connect, oltre a dettagli sulle policy AWS gestite disponibili per il servizio.

Argomenti

- Comprendi il ruolo di esecuzione del servizio
- Esempio di policy IAM per MSK Connect
- Previeni il problema della confusione tra addetti ai servizi
- AWS politiche gestite per MSK Connect
- Utilizzare ruoli collegati ai servizi per MSK Connect

Comprendi il ruolo di esecuzione del servizio

> **Note**
>
> Amazon MSK Connect non supporta l'utilizzo del ruolo collegato ai servizi come ruolo di esecuzione del servizio. È necessario creare un ruolo di esecuzione del servizio separato.

Per istruzioni su come creare un ruolo IAM personalizzato, consulta Creazione di un ruolo per delegare le autorizzazioni a un AWS servizio nella Guida per l'utente IAM.

Quando si crea un connettore con MSK Connect, è necessario specificare un ruolo AWS Identity and Access Management (IAM) da utilizzare con esso. Il ruolo di esecuzione del servizio deve disporre della seguente policy di attendibilità affinché MSK Connect lo possa assumere. Per ulteriori informazioni sulle chiavi di contesto delle condizioni in questa policy, consulta la pagina the section called “Previeni il problema della confusione tra addetti ai servizi”.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "kafkaconnect.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:kafkaconnect:us-
east-1:123456789012:connector/myConnector/abc12345-abcd-4444-a8b9-123456f513ed-2"
        }
      }
    }
  ]
}
```

Se il cluster Amazon MSK che desideri utilizzare con il connettore è un cluster che utilizza l'autenticazione IAM, devi aggiungere la seguente policy di autorizzazione al ruolo di esecuzione del servizio del connettore. Per informazioni su come trovare l'UUID del cluster e su come costruire l'argomento, consulta. ARNs the section called “Risorse relative alla politica di autorizzazione”

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:DescribeCluster"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:000000000001:cluster/
testClusterName/300d0000-0000-0005-000f-00000000000b-1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:ReadData",
                "kafka-cluster:DescribeTopic"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:123456789012:topic/
myCluster/300a0000-0000-0003-000a-00000000000b-6/__amazon_msk_connect_read"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:WriteData",
                "kafka-cluster:DescribeTopic"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:123456789012:topic/
testCluster/300f0000-0000-0008-000d-00000000000m-7/__amazon_msk_connect_write"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:CreateTopic",
                "kafka-cluster:WriteData",
```

```
                "kafka-cluster:ReadData",
                "kafka-cluster:DescribeTopic"
            ],
            "Resource": [
                "arn:aws:kafka:us-
east-1:123456789012:topic/testCluster/300f0000-0000-0008-000d-00000000000m-7/
__amazon_msk_connect_*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeGroup"
            ],
            "Resource": [
                "arn:aws:kafka:us-
east-1:123456789012:group/testCluster/300d0000-0000-0005-000f-00000000000b-1/
__amazon_msk_connect_*",
                "arn:aws:kafka:us-
east-1:123456789012:group/testCluster/300d0000-0000-0005-000f-00000000000b-1/
connect-*"
            ]
        }
    ]
}
```

A seconda del tipo di connettore, potrebbe anche essere necessario allegare al ruolo di esecuzione del servizio una politica di autorizzazioni che gli consenta di accedere alle risorse. AWS Ad esempio, se il connettore deve inviare dati a un bucket S3, il ruolo di esecuzione del servizio deve disporre di una policy di autorizzazione che conceda l'autorizzazione alla scrittura su quel bucket. A scopo di test, puoi utilizzare una delle policy IAM predefinite che forniscono l'accesso completo, come `arn:aws:iam::aws:policy/AmazonS3FullAccess`. Tuttavia, per motivi di sicurezza, si consiglia di utilizzare la politica più restrittiva che consenta al connettore di leggere dalla AWS fonte o scrivere nel AWS sink.

Esempio di policy IAM per MSK Connect

Per fornire a un utente non amministratore l'accesso completo a tutte le funzionalità di MSK Connect, collega una policy come la seguente al ruolo IAM dell'utente.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MSKConnectFullAccess",
      "Effect": "Allow",
      "Action": [
        "kafkaconnect:CreateConnector",
        "kafkaconnect:DeleteConnector",
        "kafkaconnect:DescribeConnector",
        "kafkaconnect:ListConnectors",
        "kafkaconnect:UpdateConnector",
        "kafkaconnect:CreateCustomPlugin",
        "kafkaconnect:DeleteCustomPlugin",
        "kafkaconnect:DescribeCustomPlugin",
        "kafkaconnect:ListCustomPlugins",
        "kafkaconnect:CreateWorkerConfiguration",
        "kafkaconnect:DeleteWorkerConfiguration",
        "kafkaconnect:DescribeWorkerConfiguration",
        "kafkaconnect:ListWorkerConfigurations"
      ],
      "Resource": "*"
    },
    {
      "Sid": "IAMPassRole",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/MSKConnectServiceRole",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "kafkaconnect.amazonaws.com"
        }
      }
    },
    {
      "Sid": "EC2NetworkAccess",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface",
```

```
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups"
      ],
      "Resource": "*"
    },
    {
      "Sid": "MSKClusterAccess",
      "Effect": "Allow",
      "Action": [
        "kafka:DescribeCluster",
        "kafka:DescribeClusterV2",
        "kafka:GetBootstrapBrokers"
      ],
      "Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/myCluster/"
    },
    {
      "Sid": "MSKLogGroupAccess",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams",
        "logs:DescribeLogGroups"
      ],
      "Resource": [
        "arn:aws:logs:us-east-1:123456789012:log-group:/aws/msk-connect/*"
      ]
    },
    {
      "Sid": "S3PluginAccess",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1-custom-plugins",
        "arn:aws:s3:::amzn-s3-demo-bucket1-custom-plugins/*"
      ]
    }
  ]
```

```
}
```

Previeni il problema della confusione tra addetti ai servizi

Con "confused deputy" si intende un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire una certa operazione può costringere un'entità con più privilegi a eseguire tale operazione. Nel AWS, l'impersonificazione tra servizi può causare il problema del sostituto confuso. La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua una chiamata a un altro servizio (il servizio chiamato). Il servizio chiamante può essere manipolato per utilizzare le proprie autorizzazioni e agire sulle risorse di un altro cliente, a cui normalmente non avrebbe accesso. Per evitare ciò, AWS fornisce strumenti per poterti a proteggere i tuoi dati per tutti i servizi con entità di servizio a cui è stato concesso l'accesso alle risorse del tuo account.

Ti consigliamo di utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` nelle policy delle risorse per limitare le autorizzazioni con cui MSK Connect fornisce un altro servizio alla risorsa. Se il valore `aws:SourceArn` non contiene l'ID account (ad esempio, l'AR di un bucket Amazon S3 non contiene l'ID account), è necessario utilizzare entrambe le chiavi di contesto delle condizioni globali per limitare le autorizzazioni. Se si utilizzano entrambe le chiavi di contesto delle condizioni globali e il valore `aws:SourceArn` contiene l'ID account, il valore `aws:SourceAccount` e l'account nel valore `aws:SourceArn` deve utilizzare lo stesso ID account nella stessa dichiarazione di policy. Utilizzare `aws:SourceArn` se si desidera consentire l'associazione di una sola risorsa all'accesso tra servizi. Utilizza `aws:SourceAccount` se desideri consentire l'associazione di qualsiasi risorsa in tale account all'uso tra servizi.

Nel caso di MSK Connect, il valore di `aws:SourceArn` deve essere un connettore MSK.

Il modo più efficace per proteggersi dal problema "confused deputy" è quello di usare la chiave di contesto della condizione globale `aws:SourceArn` con l'ARN completo della risorsa. Se non conosci l'ARN completo della risorsa o scegli più risorse, utilizza la chiave di contesto della condizione globale `aws:SourceArn` con caratteri jolly (*) per le parti sconosciute dell'ARN. Ad esempio, *`arn:aws:kafkaconnect:us-east-1:123456789012:connector/*`* rappresenta tutti i connettori che appartengono all'account con ID 123456789012 nella regione Stati Uniti orientali (Virginia settentrionale).

L'esempio seguente mostra il modo in cui puoi utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` in MSK Connect per prevenire il problema confused

deputy. Sostituisci *123456789012* e arn:aws:kafkaconnect: ::connector//con le tue informazioni sul connettore. *us-east-1 123456789012 my-S3-Sink-Connector abcd1234-5678-90ab-cdef-1234567890ab* Account AWS

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": " kafkaconnect.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
        "aws:SourceArn": "arn:aws:kafkaconnect:us-
east-1:123456789012:connector/my-S3-Sink-Connector/abcd1234-5678-90ab-
cdef-1234567890ab"
        }
      }
    }
  ]
}
```

AWS politiche gestite per MSK Connect

Una politica AWS gestita è una politica autonoma creata e amministrata da. AWS AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni, in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Ti consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente specifiche per i tuoi casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove operazioni API per i servizi esistenti.

Per ulteriori informazioni, consultare Policy gestite da AWSnella Guida per l'utente di IAM.

AWS politica gestita: Amazon MSKConnect ReadOnlyAccess

Questa policy concede all'utente le autorizzazioni necessarie per elencare e descrivere le risorse MSK Connect.

È possibile allegare la policy `AmazonMSKConnectReadOnlyAccess` alle identità IAM.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafkaconnect:ListConnectors",
                "kafkaconnect:ListCustomPlugins",
                "kafkaconnect:ListWorkerConfigurations"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafkaconnect:DescribeConnector"
            ],
            "Resource": [
                "arn:aws:kafkaconnect:*:*:connector/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafkaconnect:DescribeCustomPlugin"
```

```
            ],
            "Resource": [
                "arn:aws:kafkaconnect:*:*:custom-plugin/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafkaconnect:DescribeWorkerConfiguration"
            ],
            "Resource": [
                "arn:aws:kafkaconnect:*:*:worker-configuration/*"
            ]
        }
    ]
}
```

AWS politica gestita: KafkaConnectServiceRolePolicy

Questa policy concede al servizio MSK Connect le autorizzazioni necessarie per creare e gestire le interfacce di rete alle quali è assegnato il tag `AmazonMSKConnectManaged:true`. Queste interfacce di rete forniscono a MSK Connect l'accesso di rete alle risorse del tuo Amazon VPC, come un cluster Apache Kafka, un'origine o un sink.

Non puoi collegarti KafkaConnectServiceRolePolicy alle tue entità IAM. Questa policy è collegata a un ruolo collegato ai servizi che consente a MSK Connect di eseguire operazioni per tuo conto.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "ec2:CreateNetworkInterface"
   ],
   "Resource": "arn:aws:ec2:*:*:network-interface/*",
   "Condition": {
    "StringEquals": {
     "aws:RequestTag/AmazonMSKConnectManaged": "true"
```

```
    },
    "ForAllValues:StringEquals": {
     "aws:TagKeys": "AmazonMSKConnectManaged"
    }
   }
  },
  {
   "Effect": "Allow",
   "Action": [
    "ec2:CreateNetworkInterface"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:subnet/*",
    "arn:aws:ec2:*:*:security-group/*"
   ]
  },
  {
   "Effect": "Allow",
   "Action": [
    "ec2:CreateTags"
   ],
   "Resource": "arn:aws:ec2:*:*:network-interface/*",
   "Condition": {
    "StringEquals": {
     "ec2:CreateAction": "CreateNetworkInterface"
    }
   }
  },
  {
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeNetworkInterfaces",
    "ec2:CreateNetworkInterfacePermission",
    "ec2:AttachNetworkInterface",
    "ec2:DetachNetworkInterface",
    "ec2:DeleteNetworkInterface"
   ],
   "Resource": "arn:aws:ec2:*:*:network-interface/*",
   "Condition": {
    "StringEquals": {
     "ec2:ResourceTag/AmazonMSKConnectManaged": "true"
    }
   }
  }
```

```
 ]
}
```

MSK Connect aggiorna le policy AWS gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite per MSK Connect da quando questo servizio ha iniziato a tenere traccia di queste modifiche.

Modifica	Descrizione	Data
Policy di sola lettura di MSK Connect aggiornata	MSK Connect ha aggiornato la MSKConnect ReadOnlyAccess politica di Amazon per rimuovere le restrizioni sulle operazioni di pubblicazione delle offerte.	13 ottobre 2021
Inizio del tracciamento delle modifiche da parte di MSK Connect	MSK Connect ha iniziato a tenere traccia delle modifiche per le sue politiche AWS gestite.	14 settembre 2021

Utilizzare ruoli collegati ai servizi per MSK Connect

Amazon MSK Connect utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato ai servizi è un tipo di ruolo IAM univoco collegato direttamente a MSK Connect. I ruoli collegati ai servizi sono predefiniti da MSK Connect e includono tutte le autorizzazioni richieste dal servizio per chiamare altri AWS servizi per conto dell'utente.

Un ruolo collegato ai servizi semplifica la configurazione di MSK Connect perché permette di evitare l'aggiunta manuale delle autorizzazioni necessarie. MSK Connect definisce le autorizzazioni dei relativi ruoli associati ai servizi e, salvo diversamente definito, solo MSK Connect potrà assumere i propri ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consulta Servizi AWS che funzionano con IAM e cerca i servizi che riportano Sì nella colonna Ruolo associato ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni del ruolo collegato ai servizi per MSK Connect

MSK Connect utilizza il ruolo collegato al servizio denominato: AWSServiceRoleForKafkaConnectconsente ad Amazon MSK Connect di accedere alle risorse Amazon per tuo conto.

Il ruolo AWSService RoleForKafkaConnect collegato al servizio si fida che il servizio assuma il ruolo. `kafkaconnect.amazonaws.com`

Per informazioni sulla policy di autorizzazione utilizzata dal ruolo, consulta la pagina the section called “KafkaConnectServiceRolePolicy”.

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato ai servizi devi configurare le relative autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Creazione di un ruolo collegato ai servizi per MSK Connect

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando si crea un connettore nella AWS Management Console, la o l' AWS API AWS CLI, MSK Connect crea automaticamente il ruolo collegato al servizio.

Se elimini questo ruolo collegato ai servizi, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando crei un connettore, MSK Connect crea di nuovo automaticamente il ruolo collegato ai servizi per conto dell'utente.

Modifica di un ruolo collegato ai servizi per MSK Connect

MSK Connect non consente di modificare il ruolo collegato al AWSService RoleForKafkaConnect servizio. Dopo aver creato un ruolo collegato al servizio, non è possibile modificarne il nome, perché potrebbero farvi riferimento diverse entità. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta la sezione Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per MSK Connect

È possibile utilizzare la console IAM, AWS CLI o l' AWS API per eliminare manualmente il ruolo collegato al servizio. Per farlo, sarà necessario eliminare innanzitutto manualmente i connettori MSK Connect e quindi eliminare il ruolo manualmente. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi di MSK Connect

MSK Connect supporta l'utilizzo di ruoli collegati ai servizi in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta AWS Regioni ed endpoint.

Abilita l'accesso a Internet per Amazon MSK Connect

Se il tuo connettore per Amazon MSK Connect richiede l'accesso a Internet, ti consigliamo di utilizzare le seguenti impostazioni Amazon Virtual Private Cloud (VPC) per abilitare tale accesso.

- Configura il connettore con sottoreti private.
- Crea un gateway NAT pubblico o un'istanza NAT per il tuo VPC in una sottorete pubblica. Per ulteriori informazioni, consulta la pagina Connect subnet a Internet o altro VPCs utilizzando dispositivi NAT nella Guida per l'Amazon Virtual Private Cloudutente.
- Consenti il traffico in uscita dalle sottoreti private verso il gateway o l'istanza NAT.

Configurazione di un gateway NAT per Amazon MSK Connect

Nei passaggi seguenti viene illustrato come configurare un gateway NAT per abilitare l'accesso a Internet per un connettore. È necessario completare questi passaggi prima di creare un connettore in una sottorete privata.

Prerequisiti completi per la configurazione di un gateway NAT

Verifica di disporre dei seguenti elementi.

- L'ID del Amazon Virtual Private Cloud (VPC) associato al cluster. Ad esempio, vpc-123456ab.
- Le sottoreti private IDs del tuo VPC. Ad esempio, subnet-a1b2c3de, subnet-f4g5h6ij e così via. Il connettore deve essere configurato con sottoreti private.

Passaggi per abilitare l'accesso a Internet per il connettore

Abilitazione dell'accesso a Internet per il connettore

1. Apri la Amazon Virtual Private Cloud console all'indirizzo https://console.aws.amazon.com/vpc/.
2. Crea una sottorete pubblica con un nome descrittivo per il gateway NAT e prendi nota dell'ID della sottorete. Per istruzioni dettagliate, consulta la pagina Create a subnet in your VPC.
3. Crea un gateway Internet in modo che il VPC possa comunicare con Internet e prendi nota dell'ID del gateway. Collega il gateway Internet al VPC. Per istruzioni, consulta la pagina Create and attach an internet gateway.
4. Fornisci un gateway NAT pubblico in modo che gli host delle tue sottoreti private possano raggiungere la tua sottorete pubblica. Quando crei il gateway NAT, seleziona la sottorete pubblica creata in precedenza. Per istruzioni, consulta Creazione di un gateway NAT.
5. Configura le tabelle di routing. Per completare questa configurazione, occorrono in totale due tabelle di routing. Dovresti già disporre di una tabella di routing principale creata in automatico al momento della creazione del VPC. In questo passaggio creerai una tabella di routing aggiuntiva per la sottorete pubblica.

   a. Utilizza le seguenti impostazioni per modificare la tabella di routing principale del tuo VPC in modo che le sottoreti private instradino il traffico verso il tuo gateway NAT. Per le istruzioni, consulta la pagina Utilizzo delle tabelle di routing nella Guida per l'utente di Amazon Virtual Private Cloud.

      Tabella di routing MSKC privata

Proprietà	Valore
Name tag (Tag nome)	Ti consigliamo di assegnare a questa tabella di routing un nome descrittivo per facilitarne l'identificazione. Ad esempio, MSKC privata.
Sottoreti associate	Le tue sottoreti private
Un percorso per consentire l'accesso a Internet per MSK Connect	• Destinazione: 0.0.0.0/0 • Obiettivo: l'ID del gateway NAT. Ad esempio, nat-12a345bc6789efg1h.

Proprietà	Valore
Un percorso per tutto il traffico locale	• Destinazione: 10.0.0.0/16. Questo valore può variare a seconda del blocco CIDR del tuo VPC. • Obiettivo: locale

b. Segui le istruzioni riportate nella pagina Creazione di una tabella di routing personalizzata per creare una tabella di routing per la sottorete pubblica. Quando crei la tabella, inserisci un nome descrittivo nel campo Tag nome per identificare a quale sottorete è associata la tabella. Ad esempio, MSKC pubblica.

c. Configura la tua tabella di routing MSKC pubblica utilizzando le seguenti impostazioni.

Proprietà	Valore
Name tag (Tag nome)	MSKC pubblica o un altro nome descrittivo a scelta
Sottoreti associate	La tua sottorete pubblica con gateway NAT
Un percorso per consentire l'accesso a Internet per MSK Connect	• Destinazione: 0.0.0.0/0 • Obiettivo: l'ID del gateway Internet. Ad esempio, igw-1a234bc5.
Un percorso per tutto il traffico locale	• Destinazione: 10.0.0.0/16. Questo valore può variare a seconda del blocco CIDR del tuo VPC. • Obiettivo: locale

Comprendi i nomi host DNS privati

Con il supporto dei nomi host Private DNS in MSK Connect, è possibile configurare i connettori per fare riferimento a nomi di dominio pubblici o privati. Il supporto dipende dai server DNS specificati nel set di opzioni DHCP del VPC.

Un set di opzioni DHCP è un gruppo di configurazioni di rete utilizzate dalle EC2 istanze nel VPC per comunicare tramite la rete VPC. Ogni VPC ha un set di opzioni DHCP predefinito ma è possibile

creare un set di opzioni DHCP personalizzato se, ad esempio, si desidera che le istanze nel VPC utilizzino un server DNS diverso per la risoluzione dei nomi di dominio anziché il server DNS fornito da Amazon. Consulta la pagina DHCP option sets in Amazon VPC.

Prima che la funzionalità di risoluzione Private DNS fosse inclusa in MSK Connect, i connettori utilizzavano i risolutori DNS del servizio VPC per le query DNS provenienti da un connettore del cliente. I connettori non utilizzavano i server DNS definiti nei set di opzioni DHCP del VPC del cliente per la risoluzione DNS.

I connettori potevano fare riferimento solo ai nomi host nelle configurazioni dei connettori dei clienti o nei plug-in risolvibili pubblicamente. Non potevano risolvere nomi host privati definiti in una zona ospitata privatamente o utilizzare server DNS in una rete di altri clienti.

Senza Private DNS, i clienti che hanno scelto di rendere inaccessibili a Internet i propri database, data warehouse e sistemi come Secrets Manager nel proprio VPC, non potrebbero lavorare con i connettori MSK. I clienti utilizzano spesso nomi host DNS privati per conformarsi alle norme di sicurezza aziendali.

Configura un set di opzioni DHCP del VPC per il connettore

I connettori utilizzano automaticamente i server DNS definiti nel set di opzioni DHCP del VPC al momento della creazione del connettore. Prima di creare un connettore, assicurati di configurare il set di opzioni DHCP del VPC per i requisiti di risoluzione del nome host DNS del connettore.

I connettori creati prima che la funzionalità del nome host Private DNS fosse disponibile in MSK Connect continuano a utilizzare la precedente configurazione di risoluzione DNS senza che sia necessaria alcuna modifica.

Se nel connettore hai bisogno soltanto di una risoluzione dei nomi host DNS risolvibile pubblicamente, per semplificare la configurazione ti consigliamo di utilizzare il VPC predefinito del tuo account quando crei il connettore. Per ulteriori informazioni sul server DNS fornito da Amazon o Risolutore Amazon Route 53, consulta la pagina Amazon DNS Server nella Guida per l'utente di Amazon VPC.

Se devi risolvere nomi host DNS privati, assicurati che le opzioni DHCP del VPC passato durante la creazione del connettore siano impostate correttamente. Per ulteriori informazioni, consulta la pagina Work with DHCP option sets nella Guida per l'utente di Amazon VPC.

Quando configuri un set di opzioni DHCP per la risoluzione dei nomi host DNS privati, assicurati che il connettore possa raggiungere i server DNS personalizzati configurati nel set di opzioni DHCP. In caso contrario, la creazione del connettore avrà esito negativo.

Dopo aver personalizzato il set di opzioni DHCP del VPC, i connettori successivamente creati in tale VPC utilizzano i server DNS specificati nell'insieme di opzioni. Se modifichi il set di opzioni dopo aver creato un connettore, il connettore adotta le impostazioni del nuovo set di opzioni entro un paio di minuti.

Configura gli attributi DNS per il VPC

Assicurati di avere configurato correttamente gli attributi DNS del VPC come descritto nelle sezioni DNS attributes in your VPC e DNS hostnames nella Guida per l'utente di Amazon VPC.

Per informazioni sull'uso degli endpoint risolutori in entrata VPCs e in uscita per connettere altre reti al tuo VPC e lavorare con il tuo connettore, consulta la pagina Resolving DNS queries between and your network nella Guida per gli sviluppatori di Amazon Route 53.

Gestione di problemi di creazione dei connettori

Questa sezione descrive i possibili errori di creazione dei connettori associati alla risoluzione DNS e le operazioni suggerite per risolvere i problemi.

Errore	Operazione suggerita
La creazione del connettore ha esito negativo se una query di risoluzione DNS non riesce o se i server DNS non sono raggiungibili dal connettore.	Puoi visualizzare gli errori di creazione dei connettori dovuti a query di risoluzione DNS non riuscite nei CloudWatch log, se hai configurato questi log per il tuo connettore. Controlla le configurazioni del server DNS e verifica la connettività di rete ai server DNS dal connettore.
Se si modifica la configurazione dei server DNS nel set di opzioni DHCP del VPC mentre un connettore è in esecuzione, le query di risoluzio ne DNS dal connettore possono avere esito negativo. Se la risoluzione DNS non riesce,	Puoi visualizzare gli errori di creazione dei connettori dovuti a query di risoluzione DNS non riuscite nei CloudWatch log, se hai configurato questi log per il tuo connettore.

Errore	Operazione suggerita
alcune attività del connettore possono entrare in uno stato di errore.	Le attività non riuscite dovrebbero riavviarsi automaticamente per riattivare il connettore. Se ciò non accade, puoi contattare il Supporto per riavviare le attività non riuscite relative al connettore oppure puoi ricreare il connettore.

Sicurezza per MSK Connect

È possibile usare un endpoint VPC di Interface, alimentato da AWS PrivateLink, per evitare che il traffico tra Amazon VPC e Amazon MSK-Connect compatibile lasci la rete Amazon. APIs Gli endpoint VPC di interfaccia non richiedono un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione. AWS Direct Connect Per ulteriori informazioni, consulta Usa Amazon MSK APIs con endpoint VPC di interfaccia.

Registrazione per MSK Connect

MSK Connect è in grado di scrivere log eventi che è possibile utilizzare per eseguire il debug del connettore. Quando si crea un connettore, è possibile specificare nessuna, una o più delle seguenti destinazioni di log:

- Amazon CloudWatch Logs: specificate il gruppo di log a cui desiderate che MSK Connect invii gli eventi di registro del connettore. Per informazioni su come creare un gruppo di log, consulta Create a log group nella CloudWatch Logs User Guide.
- Amazon S3: specifica il bucket S3 a cui desideri che MSK Connect invii i log eventi del connettore. Per informazioni su come creare un bucket S3, consulta la pagina Creating a bucket nella Guida per l'utente di Amazon S3.
- Amazon Data Firehose: specifichi il flusso di distribuzione a cui desideri che MSK Connect invii gli eventi di registro del connettore. Per informazioni su come creare un flusso di distribuzione, consulta Creating an Amazon Data Firehose Delivery stream nella Firehose User Guide.

Per ulteriori informazioni sulla configurazione della registrazione, consulta la pagina Abilitazione della registrazione dai servizi AWS nella Guida per l'utente di Amazon CloudWatch Logs .

MSK Connect emette i seguenti tipi di log eventi:

Livello	Descrizione
INFO	Eventi di runtime di interesse all'avvio e all'arresto.
WARN	Situazioni di runtime che non sono errori ma sono indesiderate o impreviste.
FATAL	Errori gravi che causano una terminazione anticipata.
ERROR	Condizioni impreviste ed errori di runtime non fatali.

Di seguito è riportato un esempio di evento di registro inviato a CloudWatch Logs:

```
[Worker-0bb8afa0b01391c41] [2021-09-06 16:02:54,151] WARN [Producer
 clientId=producer-1] Connection to node 1 (b-1.my-test-cluster.twwhtj.c2.kafka.us-
east-1.amazonaws.com/INTERNAL_IP) could not be established. Broker may not be
 available. (org.apache.kafka.clients.NetworkClient:782)
```

Impedire la visualizzazione di segreti nei log dei connettori

> **Note**
>
> Se un plug-in non definisce i valori di configurazione sensibili come segreti, tali valori possono apparire nei log dei connettori. Kafka Connect tratta i valori di configurazione non definiti allo stesso modo di qualsiasi altro valore non crittografato.

Se il plug-in definisce una proprietà come segreta, Kafka Connect oscura il valore della proprietà nei log dei connettori. Ad esempio, i seguenti log dei connettori mostrano che se un plug-in definisce `aws.secret.key` come un tipo `PASSWORD`, il suo valore viene sostituito con **`[hidden]`**.

```
   2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b] [2022-01-11
15:18:55,150] INFO SecretsManagerConfigProviderConfig values:
   2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b] aws.access.key =
my_access_key
```

```
    2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b] aws.region = us-east-1
    2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b] aws.secret.key
= [hidden]
    2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b] secret.prefix =
    2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b] secret.ttl.ms = 300000
    2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b]
(com.github.jcustenborder.kafka.config.aws.SecretsManagerConfigProviderConfig:361)
```

Per evitare che nei file di log dei connettori appaiano dei segreti, gli sviluppatori di plug-in devono utilizzare la costante di enumerazione `ConfigDef.Type.PASSWORD` di Kafka Connect per definire le proprietà sensibili. Quando una proprietà è di tipo `ConfigDef.Type.PASSWORD`, Kafka Connect esclude il relativo valore dai log dei connettori anche se il valore viene inviato come testo non crittografato.

Monitoraggio di Amazon MSK Connect

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni di MSK Connect e delle altre AWS soluzioni. Amazon CloudWatch monitora AWS le tue risorse e le applicazioni su cui esegui AWS in tempo reale. Puoi raccogliere i parametri e tenerne traccia, creare pannelli di controllo personalizzati e impostare allarmi per inviare una notifica o intraprendere azioni quando un parametro specificato raggiunge una determinata soglia. Ad esempio, puoi tenere CloudWatch traccia dell'utilizzo della CPU o di altri parametri del connettore, in modo da aumentarne la capacità se necessario. Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.

Puoi utilizzare le seguenti operazioni API:

- `DescribeConnectorOperation`: monitora lo stato delle operazioni di aggiornamento dei connettori.
- `ListConnectorOperations`: Tieni traccia degli aggiornamenti precedenti eseguiti sul connettore.

La tabella seguente mostra le metriche inviate da MSK CloudWatch Connect all'interno della `ConnectorName` dimensione. MSK Connect fornisce queste metriche per impostazione predefinita e senza costi aggiuntivi. CloudWatch conserva queste metriche per 15 mesi, in modo da poter accedere alle informazioni storiche e avere una prospettiva migliore sulle prestazioni dei connettori. È anche possibile impostare allarmi che controllano determinate soglie e inviare notifiche o intraprendere azioni quando queste soglie vengono raggiunte. Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.

Nome parametro	Descrizione
CpuUtilization	La percentuale di utilizzo della CPU per sistema e utente.
ErroredTaskCount	Il numero di attività che sono state eseguite con errori.
MemoryUtilization	La percentuale della memoria totale su un'istanza worker, non solo la memoria heap della macchina virtuale Java (JVM) attualmente in uso. JVM in genere non restituisce la memoria al sistema operativo. Quindi, JVM heap size (MemoryUtilization) di solito inizia con una dimensione minima dell'heap che aumenta in modo incrementale fino a un massimo stabile di circa l'80-90%. L'utilizzo dell'heap JVM potrebbe aumentare o diminuire al variare dell'utilizzo effettivo della memoria da parte del connettore.
RebalanceCompletedTotal	Il numero totale di ribilanciamenti completati da questo connettore.
RebalanceTimeAvg	Il tempo medio in millisecondi impiegato dal connettore per il ribilanciamento.
RebalanceTimeMax	Il tempo massimo in millisecondi impiegato dal connettore per il ribilanciamento.
RebalanceTimeSinceLast	Il tempo in millisecondi trascorso dal momento in cui questo connettore ha completato il ribilanciamento più recente.
RunningTaskCount	Il numero di attività in esecuzione nel connettore.
SinkConsumerByteRate	Il numero medio di byte consumati al secondo dal consumatore Sink del framework Kafka

Nome parametro	Descrizione
	Connect prima dell'applicazione di qualsiasi trasformazione ai dati.
`SinkRecordReadRate`	Il numero medio al secondo di record letti dal cluster Apache Kafka o Amazon MSK.
`SinkRecordSendRate`	Il numero medio al secondo di record emessi dalle trasformazioni e inviati alla destinazione. Questo numero non include i record filtrati.
`SourceRecordPollRate`	Il numero medio al secondo di record prodotti o sottoposti a polling.
`SourceProducerByteRate`	Il numero medio di byte prodotti al secondo dal produttore del codice sorgente del framework Kafka Connect dopo l'applicazione di eventuali trasformazioni ai dati.
`SourceRecordWriteRate`	Il numero medio al secondo di record derivati dalle trasformazioni e scritti sul cluster Apache Kafka o Amazon MSK.
`TaskStartupAttemptsTotal`	Il numero totale di tentativi di avvio di attività eseguiti dal connettore. È possibile utilizzare questo parametro per identificare le anomalie nei tentativi di avvio delle attività.
`TaskStartupSuccessPercentage`	La percentuale media di attività avviate correttamente dal connettore. È possibile utilizzare questo parametro per identificare le anomalie nei tentativi di avvio delle attività.
`WorkerCount`	Il numero minimo di worker in esecuzione nel connettore.
`BytesInPerSec`	Byte di metadati trasferiti al framework Kafka Connect per la comunicazione tra i lavoratori.

Nome parametro	Descrizione
`BytesOutPerSec`	Byte di metadati trasferiti dal framework Kafka Connect per la comunicazione tra i lavoratori.

Esempi di configurazione delle risorse Amazon MSK Connect

Questa sezione include esempi per aiutarti a configurare risorse Amazon MSK Connect come connettori e provider di configurazione di terze parti comuni.

Argomenti

Configura il connettore sink Amazon S3

Questo esempio mostra come utilizzare il connettore sink Amazon S3 Confluent e AWS CLI come creare un connettore sink Amazon S3 in MSK Connect.

1. Copia il codice JSON seguente e incollalo in un nuovo file. Sostituisci le stringhe segnaposto con valori che corrispondono alla stringa di connessione dei server di bootstrap del cluster Amazon MSK e alla sottorete e al gruppo di sicurezza del cluster. IDs Per informazioni su come configurare un ruolo di esecuzione del servizio, consulta la pagina the section called “Ruoli IAM e policy”.

```
{
    "connectorConfiguration": {
        "connector.class": "io.confluent.connect.s3.S3SinkConnector",
        "s3.region": "us-east-1",
        "format.class": "io.confluent.connect.s3.format.json.JsonFormat",
        "flush.size": "1",
        "schema.compatibility": "NONE",
        "topics": "my-test-topic",
        "tasks.max": "2",
        "partitioner.class":
 "io.confluent.connect.storage.partitioner.DefaultPartitioner",
        "storage.class": "io.confluent.connect.s3.storage.S3Storage",
```

```
        "s3.bucket.name": "amzn-s3-demo-bucket"
    },
    "connectorName": "example-S3-sink-connector",
    "kafkaCluster": {
        "apacheKafkaCluster": {
            "bootstrapServers": "<cluster-bootstrap-servers-string>",
            "vpc": {
                "subnets": [
                    "<cluster-subnet-1>",
                    "<cluster-subnet-2>",
                    "<cluster-subnet-3>"
                ],
                "securityGroups": ["<cluster-security-group-id>"]
            }
        }
    },
    "capacity": {
        "provisionedCapacity": {
            "mcuCount": 2,
            "workerCount": 4
        }
    },
    "kafkaConnectVersion": "2.7.1",
    "serviceExecutionRoleArn": "<arn-of-a-role-that-msk-connect-can-assume>",
    "plugins": [
        {
            "customPlugin": {
                "customPluginArn": "<arn-of-custom-plugin-that-contains-connector-
code>",
                "revision": 1
            }
        }
    ],
    "kafkaClusterEncryptionInTransit": {"encryptionType": "PLAINTEXT"},
    "kafkaClusterClientAuthentication": {"authenticationType": "NONE"}
}
```

2. Esegui il AWS CLI comando seguente nella cartella in cui hai salvato il file JSON nel passaggio precedente.

```
aws kafkaconnect create-connector --cli-input-json file://connector-info.json
```

Di seguito è riportato un esempio dell'output che si ottiene eseguendo correttamente il comando.

```
{
    "ConnectorArn": "arn:aws:kafkaconnect:us-east-1:123450006789:connector/example-
S3-sink-connector/abc12345-abcd-4444-a8b9-123456f513ed-2",
    "ConnectorState": "CREATING",
    "ConnectorName": "example-S3-sink-connector"
}
```

Configurare il connettore del lavabo EventBridge Kafka per MSK Connect

Questo argomento mostra come configurare il connettore EventBridge Kafka sink per MSK Connect. Questo connettore consente di inviare eventi dal cluster MSK ai bus degli eventi. EventBridge Questo argomento descrive il processo di creazione delle risorse richieste e di configurazione del connettore per consentire un flusso di dati senza interruzioni tra Kafka e. EventBridge

Argomenti

- Prerequisiti
- Configurare le risorse necessarie per MSK Connect
- Crea il connettore
- Invia messaggi a Kafka

Prerequisiti

Prima di distribuire il connettore, assicurati di disporre delle seguenti risorse:

- Cluster Amazon MSK: un cluster MSK attivo per produrre e consumare messaggi Kafka.
- Amazon EventBridge Event Bus: un bus di EventBridge eventi per ricevere eventi sugli argomenti di Kafka.
- Ruoli IAM: crea ruoli IAM con le autorizzazioni necessarie per MSK Connect e il EventBridge connettore.
- Accesso a Internet pubblico da MSK Connect o da un endpoint EventBridge di interfaccia VPC creato nel VPC e nella sottorete del cluster MSK. Ciò consente di evitare di attraversare la rete Internet pubblica e di non richiedere gateway NAT.
- Una macchina client, come un' EC2 istanza Amazon o AWS CloudShell, per creare argomenti e inviare record a Kafka.

Configurare le risorse necessarie per MSK Connect

Si crea un ruolo IAM per il connettore, quindi si crea il connettore. Crei anche una EventBridge regola per filtrare gli eventi Kafka inviati al bus degli EventBridge eventi.

Argomenti

Ruolo IAM per il connettore

Il ruolo IAM associato al connettore deve disporre dell'PutEventsautorizzazione per consentire l'invio di eventi a EventBridge. Il seguente esempio di policy IAM ti concede l'autorizzazione a inviare eventi a un bus di eventi denominato. `example-event-bus` Assicurati di sostituire la risorsa ARN nell'esempio seguente con l'ARN del tuo event bus.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
      ],
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/example-event-bus"
    }
  ]
}
```

Inoltre, devi assicurarti che il tuo ruolo IAM per il connettore contenga la seguente politica di attendibilità.

JSON

```
{
```

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

Una EventBridge regola per gli eventi in arrivo

Crei regole che abbinano gli eventi in entrata ai criteri relativi ai dati degli eventi, noti come pattern di eventi. Con un pattern di eventi, è possibile definire i criteri per filtrare gli eventi in arrivo e determinare quali eventi devono attivare una particolare regola e successivamente essere indirizzati a un obiettivo designato. Il seguente esempio di schema di eventi corrisponde agli eventi di Kafka inviati al bus degli eventi. EventBridge

```
{
  "detail": {
    "topic": ["msk-eventbridge-tutorial"]
  }
}
```

Di seguito è riportato un esempio di evento inviato da Kafka all' EventBridge utilizzo del connettore Kafka sink.

```
{
  "version": "0",
  "id": "dbc1c73a-c51d-0c0e-ca61-ab9278974c57",
  "account": "123456789012",
  "time": "2025-03-26T10:15:00Z",
  "region": "us-east-1",
  "detail-type": "msk-eventbridge-tutorial",
  "source": "kafka-connect.msk-eventbridge-tutorial",
  "resources": [],
  "detail": {
    "topic": "msk-eventbridge-tutorial",
    "partition": 0,
```

```
    "offset": 0,
    "timestamp": 1742984100000,
    "timestampType": "CreateTime",
    "headers": [],
    "key": "order-1",
    "value": {
      "orderItems": [
        "item-1",
        "item-2"
      ],
      "orderCreatedTime": "Wed Mar 26 10:15:00 UTC 2025"
    }
  }
}
```

Nella EventBridge console, create una regola sul bus degli eventi utilizzando questo modello di esempio e specificate un obiettivo, ad esempio un gruppo Logs. CloudWatch La EventBridge console configurerà automaticamente la politica di accesso necessaria per il gruppo CloudWatch Logs.

Crea il connettore

Nella sezione seguente, si crea e si distribuisce il connettore EventBridge Kafka sink utilizzando il. AWS Management Console

Argomenti

- Passaggio 1: scarica il connettore
- Fase 2: creare un bucket Amazon S3
- Fase 3: Creare un plugin in MSK Connect
- Passaggio 4: creazione del connettore

Passaggio 1: scarica il connettore

Scarica l'ultimo EventBridge Connector Sink JAR dalla pagina delle GitHub versioni del connettore EventBridge Kafka. Ad esempio, per scaricare la versione v1.4.1, scegli il link al file JAR per scaricare il `kafka-eventbridge-sink-with-dependencies.jar` connettore. Quindi, salva il file in una posizione preferita sul tuo computer.

Fase 2: creare un bucket Amazon S3

1. Per archiviare il file JAR in Amazon S3 per utilizzarlo con MSK Connect, apri AWS Management Console e scegli Amazon S3.
2. Nella console Amazon S3, scegli Crea bucket e inserisci un nome di bucket univoco. Ad esempio, **`amzn-s3-demo-bucket1-eb-connector`**.
3. Scegli una regione appropriata per il tuo bucket Amazon S3. Assicurati che corrisponda alla regione in cui è distribuito il cluster MSK.
4. Per le impostazioni di Bucket, mantieni le selezioni predefinite o modificale secondo necessità.
5. Seleziona Crea bucket.
6. Carica il file JAR nel bucket Amazon S3.

Fase 3: Creare un plugin in MSK Connect

1. Aprire AWS Management Console, quindi passare a MSK Connect.
2. Nel riquadro di navigazione a sinistra, scegliete Plugin personalizzati.
3. Scegli Crea plug-in, quindi inserisci il nome del plug-in. Ad esempio, **`eventbridge-sink-plugin`**.
4. Per la posizione personalizzata del plug-in, incolla l'URL dell'oggetto S3.
5. Aggiungi una descrizione opzionale per il plugin.
6. Scegli Crea plugin.

Dopo aver creato il plug-in, è possibile utilizzarlo per configurare e distribuire il connettore EventBridge Kafka in MSK Connect.

Passaggio 4: creazione del connettore

Prima di creare il connettore, si consiglia di creare l'argomento Kafka richiesto per evitare errori nel connettore. Per creare l'argomento, utilizzate il computer client.

1. Nel riquadro sinistro della console MSK, scegli Connettori, quindi scegli Crea connettore.
2. Nell'elenco dei plugin, scegli eventbridge-sink-plugin, quindi scegli Avanti.
3. Per il nome del connettore, immettete**`EventBridgeSink`**.
4. Nell'elenco dei cluster, scegliete il vostro cluster MSK.

5. Copia la seguente configurazione per il connettore e incollala nel campo Configurazione del connettore

   Se necessario, sostituite i segnaposto nella seguente configurazione.

   - Rimuovi `aws.eventbridge.endpoint.uri` se il cluster MSK dispone di un accesso pubblico a Internet.
   - Se si utilizza PrivateLink per connettersi in modo sicuro da MSK a EventBridge, sostituire la parte DNS successiva `https://` con il nome DNS privato corretto dell'endpoint di interfaccia VPC (opzionale) per quello creato in precedenza. EventBridge
   - Sostituisci l'ARN del bus di EventBridge eventi nella seguente configurazione con l'ARN del tuo bus di eventi.
   - Aggiorna tutti i valori specifici della regione.

```
{
  "connector.class":
 "software.amazon.event.kafkaconnector.EventBridgeSinkConnector",
  "aws.eventbridge.connector.id": "msk-eventbridge-tutorial",
  "topics": "msk-eventbridge-tutorial",
  "tasks.max": "1",
  "aws.eventbridge.endpoint.uri": "https://events.us-east-1.amazonaws.com",
  "aws.eventbridge.eventbus.arn": "arn:aws:events:us-east-1:123456789012:event-bus/
example-event-bus",
  "value.converter.schemas.enable": "false",
  "value.converter": "org.apache.kafka.connect.json.JsonConverter",
  "aws.eventbridge.region": "us-east-1",
  "auto.offset.reset": "earliest",
  "key.converter": "org.apache.kafka.connect.storage.StringConverter"
}
```

   Per ulteriori informazioni sulla configurazione dei connettori, vedere. eventbridge-kafka-connector

   Se necessario, modificate le impostazioni per i lavoratori e la scalabilità automatica. Consigliamo inoltre di utilizzare l'ultima versione disponibile (consigliata) di Apache Kafka Connect dal menu a discesa. In Autorizzazioni di accesso, usa il ruolo creato in precedenza. Si consiglia inoltre di abilitare la registrazione a CloudWatch per l'osservabilità e la risoluzione dei problemi. Modifica le altre impostazioni opzionali, come i tag, in base alle tue esigenze. Quindi, distribuisci il connettore e attendi che lo stato passi allo stato In esecuzione.

Invia messaggi a Kafka

È possibile configurare le codifiche dei messaggi, come Apache Avro e JSON, specificando diversi convertitori utilizzando le impostazioni `value.converter` e, facoltativamente, disponibili in Kafka Connect. `key.converter`

connector exampleIn questo argomento è configurato per funzionare con messaggi con codifica JSON, come indicato dall'uso di for. `org.apache.kafka.connect.json.JsonConverter value converter` Quando il connettore è in stato di esecuzione, invia i record all'argomento `msk-eventbridge-tutorial` Kafka dal tuo computer client.

Usa il connettore sorgente Debezium con il provider di configurazione

Questo esempio mostra come utilizzare il plug-in del connettore Debezium MySQL con un database Amazon Aurora compatibile con MySQL come origine. In questo esempio, abbiamo anche configurato il provider open source AWS Secrets Manager Config Provider per esternalizzare le credenziali del database in AWS Secrets Manager. Per ulteriori informazioni sui provider di configurazione, consulta la pagina Tutorial: esternalizzazione di informazioni sensibili utilizzando provider di configurazione.

> **Important**
>
> Il plug-in del connettore Debezium MySQL supporta solo un'attività e non funziona con la modalità di capacità con dimensionamento automatico per Amazon MSK Connect. Dovresti invece utilizzare la modalità di capacità assegnata e impostare il valore `workerCount` su uno una nella configurazione del connettore. Per ulteriori informazioni sulle modalità di capacità di MSK Connect, consulta la pagina Comprendi la capacità dei connettori.

Prerequisiti completi per utilizzare il connettore sorgente Debezium

Il tuo connettore deve essere in grado di accedere a Internet in modo da poter interagire con servizi come AWS Secrets Manager quelli esterni al tuo. Amazon Virtual Private Cloud I passaggi descritti in questa sezione consentono di completare le seguenti attività per abilitare l'accesso a Internet.

- Configura una sottorete pubblica che ospita un gateway NAT e indirizza il traffico verso un gateway Internet nel tuo VPC.
- Crea una route predefinita che indirizza il traffico della sottorete privata verso il gateway NAT.

Per ulteriori informazioni, consulta Abilita l'accesso a Internet per Amazon MSK Connect.

Prerequisiti

Prima di abilitare l'accesso a Internet, devi disporre dei seguenti elementi:

- L'ID del Amazon Virtual Private Cloud (VPC) associato al cluster. Ad esempio, vpc-123456ab.
- Le sottoreti private IDs del tuo VPC. Ad esempio, subnet-a1b2c3de, subnet-f4g5h6ij e così via. Il connettore deve essere configurato con sottoreti private.

Abilitazione dell'accesso a Internet per il connettore

1. Apri la console all' Amazon Virtual Private Cloud indirizzo. https://console.aws.amazon.com/vpc/
2. Crea una sottorete pubblica con un nome descrittivo per il gateway NAT e prendi nota dell'ID della sottorete. Per istruzioni dettagliate, consulta la pagina Create a subnet in your VPC.
3. Crea un gateway Internet in modo che il VPC possa comunicare con Internet e prendi nota dell'ID del gateway. Collega il gateway Internet al VPC. Per istruzioni, consulta la pagina Create and attach an internet gateway.
4. Fornisci un gateway NAT pubblico in modo che gli host delle tue sottoreti private possano raggiungere la tua sottorete pubblica. Quando crei il gateway NAT, seleziona la sottorete pubblica creata in precedenza. Per istruzioni, consulta Creazione di un gateway NAT.
5. Configura le tabelle di routing. Per completare questa configurazione, occorrono in totale due tabelle di routing. Dovresti già disporre di una tabella di routing principale creata in automatico al momento della creazione del VPC. In questo passaggio creerai una tabella di routing aggiuntiva per la sottorete pubblica.

    a. Utilizza le seguenti impostazioni per modificare la tabella di routing principale del tuo VPC in modo che le sottoreti private instradino il traffico verso il tuo gateway NAT. Per le istruzioni, consulta la pagina Utilizzo delle tabelle di routing nella Guida per l'utente di Amazon Virtual Private Cloud.

    Tabella di routing MSKC privata

Proprietà	Valore
Name tag (Tag nome)	Ti consigliamo di assegnare a questa tabella di routing un nome descrittivo per

Proprietà	Valore
	facilitarne l'identificazione. Ad esempio, MSKC privata.
Sottoreti associate	Le tue sottoreti private
Un percorso per consentire l'accesso a Internet per MSK Connect	• Destinazione: 0.0.0.0/0 • Obiettivo: l'ID del gateway NAT. Ad esempio, nat-12a345bc6789efg1h.
Un percorso per tutto il traffico locale	• Destinazione: 10.0.0.0/16. Questo valore può variare a seconda del blocco CIDR del tuo VPC. • Obiettivo: locale

b. Segui le istruzioni riportate nella pagina Creazione di una tabella di routing personalizzata per creare una tabella di routing per la sottorete pubblica. Quando crei la tabella, inserisci un nome descrittivo nel campo Tag nome per identificare a quale sottorete è associata la tabella. Ad esempio, MSKC pubblica.

c. Configura la tua tabella di routing MSKC pubblica utilizzando le seguenti impostazioni.

Proprietà	Valore
Name tag (Tag nome)	MSKC pubblica o un altro nome descrittivo a scelta
Sottoreti associate	La tua sottorete pubblica con gateway NAT
Un percorso per consentire l'accesso a Internet per MSK Connect	• Destinazione: 0.0.0.0/0 • Obiettivo: l'ID del gateway Internet. Ad esempio, igw-1a234bc5.
Un percorso per tutto il traffico locale	• Destinazione: 10.0.0.0/16. Questo valore può variare a seconda del blocco CIDR del tuo VPC. • Obiettivo: locale

Ora che hai abilitato l'accesso a Internet per Amazon MSK Connect, puoi creare un connettore.

Crea un connettore sorgente Debezium

Questa procedura descrive come creare un connettore sorgente Debezium.

1. Creazione di un plug-in personalizzato

   a. Scarica il plug-in del connettore MySQL per l'ultima versione stabile dal sito Debezium. Prendi nota della versione di rilascio di Debezium che scarichi (versione 2.x o la vecchia serie 1.x). Più avanti in questa procedura, creerai un connettore basato sulla tua versione di Debezium.

   b. Scarica ed estrai AWS Secrets Manager Config Provider.

   c. Colloca i seguenti archivi nella stessa directory:

      - La cartella `debezium-connector-mysql`
      - La cartella `jcusten-border-kafka-config-provider-aws-0.1.1`

   d. Comprimi la directory che hai creato nel passaggio precedente in un file ZIP, quindi carica il file ZIP in un bucket S3. Per istruzioni, consulta la pagina Uploading objects in Amazon S3 nella Guida per l'utente di Amazon S3.

   e. Copia il codice JSON seguente e incollalo in un file. Ad esempio, `debezium-source-custom-plugin.json`. Sostituisci *<example-custom-plugin-name>* con il nome che desideri assegnare al plug-in, *<amzn-s3-demo-bucket-arn>* con l'ARN del bucket Amazon S3 in cui hai caricato il file ZIP *<file-key-of-ZIP-object>* e con la chiave del file dell'oggetto ZIP che hai caricato su S3.

```
{
    "name": "<example-custom-plugin-name>",
    "contentType": "ZIP",
    "location": {
        "s3Location": {
            "bucketArn": "<amzn-s3-demo-bucket-arn>",
            "fileKey": "<file-key-of-ZIP-object>"
        }
    }
}
```

   f. Esegui il seguente AWS CLI comando dalla cartella in cui hai salvato il file JSON per creare un plugin.

```
aws kafkaconnect create-custom-plugin --cli-input-json file://<debezium-source-custom-plugin.json>
```

Verrà visualizzato un output simile al seguente.

```
{
    "CustomPluginArn": "arn:aws:kafkaconnect:us-east-1:012345678901:custom-plugin/example-custom-plugin-name/abcd1234-a0b0-1234-c1-12345678abcd-1",
    "CustomPluginState": "CREATING",
    "Name": "example-custom-plugin-name",
    "Revision": 1
}
```

g. Esegui il comando seguente per verificare lo stato del plug-in. Lo stato del cluster dovrebbe passare da `CREATING` a `ACTIVE`. Sostituisci il segnaposto ARN con l'ARN ottenuto nell'output del comando precedente.

```
aws kafkaconnect describe-custom-plugin --custom-plugin-arn "<arn-of-your-custom-plugin>"
```

2. Configura AWS Secrets Manager e crea un segreto per le credenziali del tuo database

   a. Apri la console Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.

   b. Crea un nuovo segreto per archiviare le credenziali di accesso al database. Per le istruzioni, consulta la pagina Create a secret nella Guida per l'utente di AWS Secrets Manager.

   c. Copia l'ARN del segreto.

   d. Aggiungi le autorizzazioni di Secrets Manager dalla seguente policy di esempio al tuo Comprendi il ruolo di esecuzione del servizio. Sostituisci *<arn:aws:secretsmanager:us-east-1:123456789000:secret:MySecret-1234>* con l'ARN del tuo segreto.

   JSON

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

```
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetResourcePolicy",
        "secretsmanager:GetSecretValue",
        "secretsmanager:DescribeSecret",
        "secretsmanager:ListSecretVersionIds"
      ],
      "Resource": [
      "arn:aws:secretsmanager:us-
east-1:123456789012:secret:MySecret-1234"
      ]
    }
  ]
}
```

Per istruzioni sull'aggiunta di autorizzazioni IAM, consulta la pagina Adding and removing IAM identity permissions nella Guida per l'utente di IAM.

3. Creazione di una configurazione del worker personalizzata con informazioni sul proprio provider di configurazione

   a. Copia le seguenti proprietà di configurazione del worker in un file, sostituendo le stringhe segnaposto con valori che corrispondono al tuo scenario. Per ulteriori informazioni sulle proprietà di configurazione per il provider di configurazione di AWS Secrets Manager Config, SecretsManagerConfigProviderconsultate la documentazione del plugin.

   ```
   key.converter=<org.apache.kafka.connect.storage.StringConverter>
   value.converter=<org.apache.kafka.connect.storage.StringConverter>
   config.providers.secretManager.class=com.github.jcustenborder.kafka.config.aws.SecretsM
   config.providers=secretManager
   config.providers.secretManager.param.aws.region=<us-east-1>
   ```

   b. Esegui il AWS CLI comando seguente per creare la tua configurazione di lavoro personalizzata.

   Sostituisci i valori seguenti:

   - *<my-worker-config-name>*- un nome descrittivo per la configurazione personalizzata del lavoratore
   - *<encoded-properties-file-content-string>*- una versione con codifica base64 delle proprietà di testo in chiaro copiate nel passaggio precedente

```
aws kafkaconnect create-worker-configuration --name <my-worker-config-name> --
properties-file-content <encoded-properties-file-content-string>
```

4. Creazione di un connettore

   a. Copia il codice JSON seguente, che corrisponde alla tua versione di Debezium (2.x o 1.x), e incollalo in un nuovo file. Sostituisci le stringhe *<placeholder>* con valori che corrispondono al tuo scenario. Per informazioni su come configurare un ruolo di esecuzione del servizio, consulta la pagina the section called “Ruoli IAM e policy”.

      Nota che la configurazione utilizza variabili come `${secretManager:MySecret-1234:dbusername}` anziché testo non crittografato per specificare le credenziali del database. Sostituisci *MySecret-1234* con il nome del tuo segreto, poi includi il nome della chiave che desideri recuperare. È inoltre necessario sostituire *<arn-of-config-provider-worker-configuration>* con l'ARN della configurazione del worker personalizzata.

      Debezium 2.x

      Per le versioni di Debezium 2.x, copia il codice JSON seguente e incollalo in un nuovo file. Sostituisci le stringhe *<placeholder>* con valori che corrispondono al tuo scenario.

```
{
 "connectorConfiguration": {
  "connector.class": "io.debezium.connector.mysql.MySqlConnector",
  "tasks.max": "1",
  "database.hostname": "<aurora-database-writer-instance-endpoint>",
  "database.port": "3306",
  "database.user": "<${secretManager:MySecret-1234:dbusername}>",
  "database.password": "<${secretManager:MySecret-1234:dbpassword}>",
  "database.server.id": "123456",
  "database.include.list": "<list-of-databases-hosted-by-specified-server>",
  "topic.prefix": "<logical-name-of-database-server>",
  "schema.history.internal.kafka.topic": "<kafka-topic-used-by-debezium-to-
track-schema-changes>",
  "schema.history.internal.kafka.bootstrap.servers": "<cluster-bootstrap-
servers-string>",
  "schema.history.internal.consumer.security.protocol": "SASL_SSL",
  "schema.history.internal.consumer.sasl.mechanism": "AWS_MSK_IAM",
```

```
  "schema.history.internal.consumer.sasl.jaas.config":
 "software.amazon.msk.auth.iam.IAMLoginModule required;",
  "schema.history.internal.consumer.sasl.client.callback.handler.class":
 "software.amazon.msk.auth.iam.IAMClientCallbackHandler",
  "schema.history.internal.producer.security.protocol": "SASL_SSL",
  "schema.history.internal.producer.sasl.mechanism": "AWS_MSK_IAM",
  "schema.history.internal.producer.sasl.jaas.config":
 "software.amazon.msk.auth.iam.IAMLoginModule required;",
  "schema.history.internal.producer.sasl.client.callback.handler.class":
 "software.amazon.msk.auth.iam.IAMClientCallbackHandler",
  "include.schema.changes": "true"
 },
 "connectorName": "example-Debezium-source-connector",
 "kafkaCluster": {
  "apacheKafkaCluster": {
   "bootstrapServers": "<cluster-bootstrap-servers-string>",
   "vpc": {
    "subnets": [
     "<cluster-subnet-1>",
     "<cluster-subnet-2>",
     "<cluster-subnet-3>"
    ],
    "securityGroups": ["<id-of-cluster-security-group>"]
   }
  }
 },
 "capacity": {
  "provisionedCapacity": {
   "mcuCount": 2,
   "workerCount": 1
  }
 },
 "kafkaConnectVersion": "2.7.1",
 "serviceExecutionRoleArn": "<arn-of-service-execution-role-that-msk-
connect-can-assume>",
 "plugins": [{
  "customPlugin": {
   "customPluginArn": "<arn-of-msk-connect-plugin-that-contains-connector-
code>",
   "revision": 1
  }
 }],
 "kafkaClusterEncryptionInTransit": {
  "encryptionType": "TLS"
```

```
 },
 "kafkaClusterClientAuthentication": {
  "authenticationType": "IAM"
 },
 "workerConfiguration": {
  "workerConfigurationArn": "<arn-of-config-provider-worker-configuration>",
  "revision": 1
 }
}
```

Debezium 1.x

Per le versioni di Debezium 1.x, copia il codice JSON seguente e incollalo in un nuovo file. Sostituisci le stringhe *<placeholder>* con valori che corrispondono al tuo scenario.

```
{
 "connectorConfiguration": {
  "connector.class": "io.debezium.connector.mysql.MySqlConnector",
  "tasks.max": "1",
  "database.hostname": "<aurora-database-writer-instance-endpoint>",
  "database.port": "3306",
  "database.user": "<${secretManager:MySecret-1234:dbusername}>",
  "database.password": "<${secretManager:MySecret-1234:dbpassword}>",
  "database.server.id": "123456",
  "database.server.name": "<logical-name-of-database-server>",
  "database.include.list": "<list-of-databases-hosted-by-specified-server>",
  "database.history.kafka.topic": "<kafka-topic-used-by-debezium-to-track-
schema-changes>",
  "database.history.kafka.bootstrap.servers": "<cluster-bootstrap-servers-
string>",
  "database.history.consumer.security.protocol": "SASL_SSL",
  "database.history.consumer.sasl.mechanism": "AWS_MSK_IAM",
  "database.history.consumer.sasl.jaas.config":
 "software.amazon.msk.auth.iam.IAMLoginModule required;",
  "database.history.consumer.sasl.client.callback.handler.class":
 "software.amazon.msk.auth.iam.IAMClientCallbackHandler",
  "database.history.producer.security.protocol": "SASL_SSL",
  "database.history.producer.sasl.mechanism": "AWS_MSK_IAM",
  "database.history.producer.sasl.jaas.config":
 "software.amazon.msk.auth.iam.IAMLoginModule required;",
  "database.history.producer.sasl.client.callback.handler.class":
 "software.amazon.msk.auth.iam.IAMClientCallbackHandler",
  "include.schema.changes": "true"
```

```
 },
 "connectorName": "example-Debezium-source-connector",
 "kafkaCluster": {
  "apacheKafkaCluster": {
   "bootstrapServers": "<cluster-bootstrap-servers-string>",
   "vpc": {
    "subnets": [
     "<cluster-subnet-1>",
     "<cluster-subnet-2>",
     "<cluster-subnet-3>"
    ],
    "securityGroups": ["<id-of-cluster-security-group>"]
   }
  }
 },
 "capacity": {
  "provisionedCapacity": {
   "mcuCount": 2,
   "workerCount": 1
  }
 },
 "kafkaConnectVersion": "2.7.1",
 "serviceExecutionRoleArn": "<arn-of-service-execution-role-that-msk-
connect-can-assume>",
 "plugins": [{
  "customPlugin": {
   "customPluginArn": "<arn-of-msk-connect-plugin-that-contains-connector-
code>",
   "revision": 1
  }
 }],
 "kafkaClusterEncryptionInTransit": {
  "encryptionType": "TLS"
 },
 "kafkaClusterClientAuthentication": {
  "authenticationType": "IAM"
 },
 "workerConfiguration": {
  "workerConfigurationArn": "<arn-of-config-provider-worker-configuration>",
  "revision": 1
 }
}
```

b. Esegui il AWS CLI comando seguente nella cartella in cui hai salvato il file JSON nel passaggio precedente.

```
aws kafkaconnect create-connector --cli-input-json file://connector-info.json
```

Di seguito è riportato un esempio dell'output che si ottiene eseguendo correttamente il comando.

```
{
    "ConnectorArn": "arn:aws:kafkaconnect:us-east-1:123450006789:connector/
example-Debezium-source-connector/abc12345-abcd-4444-a8b9-123456f513ed-2",
    "ConnectorState": "CREATING",
    "ConnectorName": "example-Debezium-source-connector"
}
```

Aggiorna la configurazione di un connettore Debezium

Per aggiornare la configurazione del connettore Debezium, segui questi passaggi:

1. Copia il seguente codice JSON e incollalo in un nuovo file. Sostituisci le stringhe `<placeholder>` con valori che corrispondono al tuo scenario.

```
{
   "connectorArn": <connector_arn>,
   "connectorConfiguration": <new_configuration_in_json>,
   "currentVersion": <current_version>
}
```

2. Esegui il AWS CLI comando seguente nella cartella in cui hai salvato il file JSON nel passaggio precedente.

```
aws kafkaconnect update-connector --cli-input-json file://connector-info.json
```

Di seguito è riportato un esempio dell'output ottenuto quando si esegue correttamente il comando.

```
{
    "connectorArn": "arn:aws:kafkaconnect:us-east-1:123450006789:connector/example-
Debezium-source-connector/abc12345-abcd-4444-a8b9-123456f513ed-2",
```

```
    "connectorOperationArn": "arn:aws:kafkaconnect:us-
east-1:123450006789:connector-operation/example-Debezium-source-connector/abc12345-
abcd-4444-a8b9-123456f513ed-2/41b6ad56-3184-479b-850a-a8bedd5a02f3",
    "connectorState": "UPDATING"
}
```

3. È ora possibile eseguire il comando seguente per monitorare lo stato corrente dell'operazione:

```
aws kafkaconnect describe-connector-operation --connector-operation-arn
 <operation_arn>
```

Per un esempio di connettore Debezium con i passaggi dettagliati, consulta la pagina Introducing Amazon MSK Connect - Stream Data to and from Your Apache Kafka Clusters Using Managed Connectors.

Migrazione ad Amazon MSK Connect

Questa sezione descrive come migrare l'applicazione del connettore Apache Kafka su Amazon Managed Streaming for Apache Kafka Connect (Amazon MSK Connect). Per ulteriori informazioni sui vantaggi della migrazione ad Amazon MSK Connect, consulta. ???

Questa sezione descrive anche gli argomenti di gestione dello stato utilizzati da Kafka Connect e Amazon MSK Connect e descrive le procedure per la migrazione dei connettori source e sink.

Comprendi gli argomenti interni utilizzati da Kafka Connect

Un'applicazione Apache Kafka Connect in esecuzione in modalità distribuita memorizza il proprio stato utilizzando argomenti interni nel cluster Kafka e l'appartenenza ai gruppi. I seguenti sono i valori di configurazione che corrispondono agli argomenti interni utilizzati per le applicazioni Kafka Connect:

- Argomento di configurazione, specificato tramite `config.storage.topic`

  Nell'argomento di configurazione, Kafka Connect memorizza la configurazione di tutti i connettori e le attività avviate dagli utenti. Ogni volta che gli utenti aggiornano la configurazione di un connettore o quando un connettore richiede una riconfigurazione (ad esempio, il connettore rileva che può avviare più attività), viene emesso un record su questo argomento. Questo argomento è abilitato alla compattazione, quindi mantiene sempre l'ultimo stato per ogni entità.
- Argomento Offsets, specificato tramite `offset.storage.topic`

Nell'argomento offset, Kafka Connect memorizza gli offset dei connettori sorgente. Come l'argomento sulla configurazione, l'argomento offset è abilitato alla compattazione. Questo argomento viene utilizzato per scrivere le posizioni di origine solo per i connettori di origine che producono dati a Kafka da sistemi esterni. I connettori Sink, che leggono i dati da Kafka e li inviano a sistemi esterni, memorizzano i dati di consumo utilizzando i normali gruppi di consumatori Kafka.

- Argomento relativo allo stato, specificato tramite `status.storage.topic`

  Nell'argomento relativo allo stato, Kafka Connect memorizza lo stato corrente dei connettori e delle attività. Questo argomento viene utilizzato come punto centrale per i dati richiesti dagli utenti dell'API REST. Questo argomento consente agli utenti di interrogare qualsiasi worker e ottenere comunque lo stato di tutti i plugin in esecuzione. Come gli argomenti di configurazione e offset, anche l'argomento status è abilitato alla compattazione.

Oltre a questi argomenti, Kafka Connect fa ampio uso dell'API di appartenenza ai gruppi di Kafka. I gruppi prendono il nome dal nome del connettore. Ad esempio, per un connettore denominato file-sink, il gruppo viene denominato. connect-file-sink Ogni consumatore del gruppo fornisce i record relativi a una singola attività. Questi gruppi e i relativi offset possono essere recuperati utilizzando i normali strumenti dei gruppi di consumatori, come. `Kafka-consumer-group.sh` Per ogni connettore sink, il runtime Connect esegue un normale gruppo di consumatori che estrae i record da Kafka.

Gestione dello stato delle applicazioni Amazon MSK Connect

Per impostazione predefinita, Amazon MSK Connect crea tre argomenti separati nel cluster Kafka per ogni connettore Amazon MSK per memorizzare la configurazione, l'offset e lo stato del connettore. I nomi degli argomenti predefiniti sono strutturati come segue:

- *`connector-name`*__msk_connect_configs_ _ *`connector-id`*
- __msk_connect_status_ _ *`connector-name connector-id`*
- __msk_connect_offsets_ _ *`connector-name connector-id`*

> **Note**
>
> Per garantire la continuità degli offset tra i connettori di origine, puoi utilizzare un argomento di archiviazione degli offset a tua scelta, anziché l'argomento predefinito. La definizione di un argomento di archiviazione degli offset consente di eseguire attività come la creazione di

un connettore di origine che riprenda la lettura dall'ultimo offset di un connettore precedente. Per definire un argomento di archiviazione degli offset, è necessario fornire un valore per la offset.storage.topicproprietà nella configurazione del worker di Amazon MSK Connect.

Migrazione dei connettori di origine ad Amazon MSK Connect

I connettori di origine sono applicazioni Apache Kafka Connect che importano record da sistemi esterni in Kafka. Questa sezione descrive il processo di migrazione delle applicazioni del connettore di origine Apache Kafka Connect che eseguono cluster Kafka Connect locali o autogestiti in esecuzione su Amazon MSK Connect. AWS

L'applicazione Kafka Connect source connector memorizza gli offset in un argomento denominato con il valore impostato per la proprietà config. `offset.storage.topic` Di seguito sono riportati alcuni esempi di messaggi di offset per un connettore JDBC che esegue due attività che importano dati da due tabelle diverse denominate e. `movies` `shows` La riga più recente importata dai film da tavolo ha un ID primario di. `18343` La riga più recente importata dalla tabella shows ha un ID primario di`732`.

```
["jdbcsource",{"protocol":"1","table":"sample.movies"}] {"incrementing":18343}
["jdbcsource",{"protocol":"1","table":"sample.shows"}] {"incrementing":732}
```

Per eseguire la migrazione dei connettori di origine ad Amazon MSK Connect, procedere come segue:

1. Crea un plug-in personalizzato Amazon MSK Connect estraendo le librerie di connettori dal tuo cluster Kafka Connect locale o autogestito.
2. Crea le proprietà dei lavoratori Amazon MSK Connect e imposta le proprietà `key.converter` e `offset.storage.topic` gli stessi valori impostati per il connettore Kafka in esecuzione nel tuo cluster Kafka Connect esistente. `value.converter`
3. Metti in pausa l'applicazione del connettore sul cluster esistente effettuando una `PUT /connectors/`*`connector-name`*`/pause` richiesta sul cluster Kafka Connect esistente.
4. Assicurati che tutte le attività dell'applicazione del connettore siano completamente interrotte. È possibile interrompere le attività effettuando una `GET /connectors/`*`connector-name`*`/status` richiesta sul cluster Kafka Connect esistente o consumando i messaggi dal nome dell'argomento impostato per la proprietà. `status.storage.topic`

5. Ottieni la configurazione del connettore dal cluster esistente. È possibile ottenere la configurazione del connettore effettuando una `GET /connectors/connector-name/config/` richiesta sul cluster esistente o utilizzando i messaggi dal nome dell'argomento impostato per la proprietà`config.storage.topic`.
6. Crea un nuovo Amazon MSK Connector con lo stesso nome di un cluster esistente. Crea questo connettore utilizzando il plug-in personalizzato del connettore che hai creato nel passaggio 1, le proprietà del worker che hai creato nel passaggio 2 e la configurazione del connettore che hai estratto nel passaggio 5.
7. Quando lo stato di Amazon MSK Connector è `active` impostato su, visualizza i log per verificare che il connettore abbia iniziato a importare dati dal sistema di origine.
8. Elimina il connettore nel cluster esistente effettuando una richiesta. `DELETE /connectors/connector-name`

Esegui la migrazione dei connettori sink ad Amazon MSK Connect

I connettori Sink sono applicazioni Apache Kafka Connect che esportano dati da Kafka a sistemi esterni. Questa sezione descrive il processo di migrazione delle applicazioni sink connector di Apache Kafka Connect che eseguono cluster Kafka Connect locali o autogestiti in esecuzione su Amazon MSK Connect. AWS

I connettori sink Kafka Connect utilizzano l'API di appartenenza al gruppo Kafka e memorizzano gli offset negli stessi `__consumer_offset` argomenti di una tipica applicazione per consumatori. Questo comportamento semplifica la migrazione del connettore sink da un cluster autogestito ad Amazon MSK Connect.

Per eseguire la migrazione dei connettori del sink ad Amazon MSK Connect, procedere come segue:

1. Crea un plug-in personalizzato Amazon MSK Connect estraendo le librerie di connettori dal tuo cluster Kafka Connect locale o autogestito.
2. Crea le proprietà dei lavoratori Amazon MSK Connect e imposta le proprietà `key.converter` e `value.converter` gli stessi valori impostati per il connettore Kafka in esecuzione nel tuo cluster Kafka Connect esistente.
3. Metti in pausa l'applicazione del connettore sul cluster esistente effettuando una `PUT /connectors/connector-name/pause` richiesta sul cluster Kafka Connect esistente.
4. Assicurati che tutte le attività dell'applicazione del connettore siano completamente interrotte. È possibile interrompere le attività effettuando una `GET /connectors/connector-name/status`

richiesta sul cluster Kafka Connect esistente o consumando i messaggi dal nome dell'argomento impostato per la proprietà. `status.storage.topic`

5. Ottieni la configurazione del connettore dal cluster esistente. È possibile ottenere la configurazione del connettore effettuando una `GET /connectors/`*`connector-name`*`/config` richiesta sul cluster esistente o utilizzando i messaggi dal nome dell'argomento impostato per la proprietà`config.storage.topic`.
6. Crea un nuovo Amazon MSK Connector con lo stesso nome del cluster esistente. Crea questo connettore utilizzando il plug-in personalizzato del connettore che hai creato nel passaggio 1, le proprietà del worker che hai creato nel passaggio 2 e la configurazione del connettore che hai estratto nel passaggio 5.
7. Quando lo stato di Amazon MSK Connector è `active` impostato su, visualizza i log per verificare che il connettore abbia iniziato a importare dati dal sistema di origine.
8. Elimina il connettore nel cluster esistente effettuando una richiesta. `DELETE /connectors/`*`connector-name`*

Risolvi i problemi in Amazon MSK Connect

Le seguenti informazioni agevolano la risoluzione dei problemi che si potrebbero verificare durante l'utilizzo di MSK Connect. Puoi anche pubblicare il problema nel AWS re:Post.

Il connettore non è in grado di accedere alle risorse ospitate sulla rete Internet pubblica

Consulta la sezione Abilitazione dell'accesso a Internet per Amazon MSK Connect.

Il numero di attività in esecuzione del connettore non corrisponde al numero di attività specificato in tasks.max

Ecco alcuni motivi per cui un connettore può utilizzare un numero inferiore di attività rispetto alla configurazione tasks.max specificata:

- Alcune implementazioni di connettori limitano il numero di attività che è possibile utilizzare. Ad esempio, il connettore Debezium per MySQL è limitato all'utilizzo di una singola attività.
- Quando si utilizza la modalità di capacità con scalabilità automatica, Amazon MSK Connect sostituisce la proprietà tasks.max del connettore con un valore proporzionale al numero di lavoratori in esecuzione nel connettore e al numero di lavoratori per lavoratore. MCUs

- Per i connettori sink, il livello di parallelismo (numero di attività) non può essere superiore al numero di partizioni di argomento. Sebbene sia possibile impostare tasks.max su un valore maggiore, una singola partizione non viene mai elaborata da più di una singola attività alla volta.
- In Kafka Connect 2.7.x, l'assegnatore di partizioni dei consumatori predefinito è `RangeAssignor`. Il comportamento di questo assegnatore è quello di assegnare la prima partizione di ogni argomento a un singolo consumatore, la seconda partizione di ogni argomento a un singolo consumatore ecc. Ciò significa che il numero massimo di attività attive utilizzate da un connettore sink tramite `RangeAssignor` è uguale al numero massimo di partizioni utilizzate in ogni singolo argomento utilizzato. Se ciò non funziona per il tuo caso d'uso, dovresti creare una configurazione del worker in cui la proprietà `consumer.partition.assignment.strategy` sia impostata su un assegnatore di partizioni dei consumatori più adatto. Vedi Interfaccia Kafka 2.7: tutte le classi di implementazione conosciute. ConsumerPartitionAssignor

Cos'è il replicatore Amazon MSK?

Amazon MSK Replicator è una funzionalità di Amazon MSK che consente di replicare in modo affidabile i dati tra cluster Amazon MSK diversi o uguali. Regione AWS Tuttavia, sia il cluster di origine che quello di destinazione devono trovarsi nello stesso ambiente. Account AWS Con il replicatore MSK, è possibile creare facilmente applicazioni di streaming resilienti a livello regionale per una maggiore disponibilità e continuità aziendale. Il replicatore MSK fornisce la replica asincrona automatica tra i cluster MSK, eliminando la necessità di scrivere codice personalizzato, gestire l'infrastruttura o configurare reti tra regioni.

Il replicatore MSK dimensiona automaticamente le risorse sottostanti in modo da poter replicare i dati on demand senza dover monitorare o dimensionare la capacità. MSK Replicator replica anche i metadati Kafka necessari, tra cui le configurazioni degli argomenti, le liste di controllo degli accessi () e gli offset dei gruppi di consumatori. ACLs Se si verifica un evento imprevisto in una regione, è possibile eseguire il failover nell'altra regione e riprendere l'elaborazione senza interruzioni. AWS

Il replicatore MSK supporta sia la replica tra regioni (CRR) sia la replica nella stessa regione (SRR). Nella replica tra regioni, i cluster MSK di origine e di destinazione si trovano in regioni diverse. AWS Nella replica nella stessa regione, i cluster MSK di origine e di destinazione si trovano nella stessa regione. AWS È necessario creare cluster MSK di origine e di destinazione prima di poterli utilizzare con il replicatore MSK.

> **Note**
>
> MSK Replicator supporta le seguenti AWS regioni: Stati Uniti orientali (us-east-1, Virginia settentrionale); Stati Uniti orientali (us-east-2, Ohio); Stati Uniti occidentali (us-west-2, Oregon); Europa (eu-west-1, Irlanda); Europa (eu-central-1, Francoforte); Asia Pacifico (ap-southeast-1, Singapore); Asia Pacifico (ap-southeast-2, Sydney), Europa (eu-north-1, Stoccolma), Asia Pacifico (ap-south-1, Mumbai), Europa (eu-west-3, Parigi), Sud America (sa-east-1, San Paolo), Asia Pacifico (ap-northeast-2, Seoul), Europa (eu-west-2, Londra), Asia Pacifico (ap-northeast-1, Tokyo), Stati Uniti occidentali (us-west-1, California settentrionale), Canada (ca-central-1, centrale).

Ecco alcuni usi comuni di Amazon MSK Replicator.

- Crea applicazioni di streaming tra regioni: crea applicazioni di streaming ad alta disponibilità e tolleranti ai guasti per conseguire una maggiore resilienza senza necessità di configurare soluzioni personalizzate.
- Fornisci un accesso ai dati a bassa latenza: offri un accesso ai dati con latenza inferiore ai consumatori ubicati in aree geografiche diverse.
- Distribuisci i dati ai tuoi partner: copia i dati da un cluster Apache Kafka a più cluster Apache Kafka, in modo che diversi abbiano le proprie copie dei dati. teams/partners
- Aggrega i dati per l'analisi: copia i dati da più cluster Apache Kafka in un unico cluster per generare facilmente approfondimenti su dati aggregati in tempo reale.
- Scrivi localmente, accedi ai tuoi dati a livello globale: configura la replica multiattiva per propagare automaticamente le scritture eseguite in una AWS regione ad altre regioni per fornire dati a latenza e costi inferiori.

Come funziona il replicatore Amazon MSK

Per iniziare a utilizzare MSK Replicator, è necessario creare un nuovo replicatore nella regione del cluster di destinazione. AWS MSK Replicator copia automaticamente tutti i dati dal cluster nella AWS regione primaria denominata origine nel cluster nella regione di destinazione denominata destinazione. I cluster di origine e di destinazione possono trovarsi nella stessa regione o in regioni diverse. AWS Se il cluster di destinazione non esiste ancora, devi crearlo.

Quando si crea un replicatore, MSK Replicator distribuisce tutte le risorse necessarie nella AWS regione del cluster di destinazione per ottimizzare la latenza di replica dei dati. La latenza di replica varia in base a molti fattori, tra cui la distanza di rete tra le AWS regioni dei cluster MSK, la capacità di throughput dei cluster di origine e di destinazione e il numero di partizioni sui cluster di origine e di destinazione. Il replicatore MSK dimensiona automaticamente le risorse sottostanti in modo da poter replicare i dati on demand senza dover monitorare o dimensionare la capacità.

Replica dei dati

Per impostazione predefinita, MSK Replicator copia tutti i dati in modo asincrono dall'ultimo offset nelle partizioni tematiche del cluster di origine nel cluster di destinazione. Se l'impostazione «Rileva e copia nuovi argomenti» è attivata, MSK Replicator rileva e copia automaticamente nuovi argomenti o partizioni di argomenti nel cluster di destinazione. Tuttavia, il Replicator potrebbe impiegare fino a 30 secondi per rilevare e creare nuovi argomenti o partizioni di argomenti nel cluster di destinazione. Tutti i messaggi inviati all'argomento di origine prima della creazione dell'argomento nel cluster di

destinazione non verranno replicati. In alternativa, è possibile configurare il Replicator durante la creazione per avviare la replica dal primo offset nelle partizioni degli argomenti del cluster di origine se si desidera replicare i messaggi esistenti sui propri argomenti nel cluster di destinazione.

MSK Replicator non archivia i dati. I dati vengono utilizzati dal cluster di origine, inseriti nel buffer in memoria e scritti nel cluster di destinazione. Il buffer viene cancellato automaticamente quando i dati vengono scritti correttamente o hanno esito negativo dopo nuovi tentativi. Tutte le comunicazioni e i dati tra MSK Replicator e i cluster sono sempre crittografati durante il transito. Tutte le chiamate API MSK Replicator, ad esempio`DescribeClusterV2`, vengono acquisite in. `CreateTopic` `DescribeTopicDynamicConfiguration` AWS CloudTrail Anche i log del vostro broker MSK rifletteranno la stessa cosa.

MSK Replicator crea argomenti nel cluster di destinazione con un fattore di replica pari a 3. Se necessario, è possibile modificare il fattore di replica direttamente sul cluster di destinazione.

Replica dei metadati

MSK Replicator supporta anche la copia dei metadati dal cluster di origine al cluster di destinazione. I metadati includono la configurazione degli argomenti, le liste di controllo degli accessi (ACLs) e gli offset dei gruppi di consumatori. Come la replica dei dati, anche la replica dei metadati avviene in modo asincrono. Per prestazioni migliori, MSK Replicator dà priorità alla replica dei dati rispetto alla replica dei metadati.

La tabella seguente è un elenco di elenchi di controllo degli accessi () copiati da MSK Replicator. ACLs

Operazione	Ricerca	APIs consentito
Alter	Argomento	CreatePartitions
AlterConfigs	Argomento	AlterConfigs
Crea	Argomento	CreateTopics, Metadati
Eliminazione	Argomento	DeleteRecords, DeleteTopics
Describe	Argomento	ListOffsets, Metadati, OffsetFetch OffsetFor LeaderEpoch

Operazione	Ricerca	APIs consentito
DescribeConfigs	Argomento	DescribeConfigs
Lettura	Argomento	Recupera,, OffsetCommit TxnOffsetCommit
Scrivi (solo nega)	Argomento	Produrre, AddPartitionsToTxn

MSK Replicator copia il tipo di pattern LITERAL ACLs solo per il tipo di risorsa Topic. Il tipo di pattern PREFISSO ACLs e l'altro tipo di risorsa non vengono copiati. ACLs Inoltre, MSK Replicator non esegue l'eliminazione ACLs sul cluster di destinazione. Se si elimina un ACL sul cluster di origine, è necessario eliminarlo contemporaneamente anche sul cluster di destinazione. Per maggiori dettagli sulle ACLs risorse, il pattern e le operazioni di Kafka, consulta https://kafka.apache.org/documentation/#security_authz_cli.

MSK Replicator replica solo Kafka ACLs, che il controllo degli accessi IAM non utilizza. Se i clienti utilizzano il controllo degli accessi IAM ai cluster MSK, è necessario configurare le politiche IAM pertinenti anche sul cluster di destinazione per un failover senza interruzioni. read/write Questo vale anche per le configurazioni di replica dei nomi degli argomenti con prefisso e identico.

Nell'ambito della sincronizzazione degli offset dei gruppi di consumatori, MSK Replicator effettua l'ottimizzazione per i consumatori del cluster di origine che leggono da una posizione più vicina alla fine del flusso (partizione di fine argomento). Se i gruppi di consumatori sono in ritardo rispetto al cluster di origine, è possibile riscontrare un ritardo maggiore per tali gruppi di consumatori sul cluster di destinazione rispetto a quello di origine. Ciò significa che, dopo il failover sul cluster di destinazione, i consumatori rielaboreranno più messaggi duplicati. Per ridurre questo ritardo, i tuoi utenti del cluster di origine dovrebbero recuperare il ritardo e iniziare a consumare dall'estremità dello stream (fine della partizione dell'argomento). Man mano che i consumatori recuperano il ritardo, MSK Replicator ridurrà automaticamente il ritardo.

Configurazione del nome dell'argomento

MSK Replicator dispone di due modalità di configurazione dei nomi degli argomenti: replica dei nomi degli argomenti con prefisso (impostazione predefinita) o replica identica dei nomi degli argomenti.

Replica dei nomi degli argomenti con prefisso

Per impostazione predefinita, MSK Replicator crea nuovi argomenti nel cluster di destinazione con un prefisso generato automaticamente aggiunto al nome dell'argomento del cluster di origine, ad esempio. `<sourceKafkaClusterAlias>.topic` Questo serve a distinguere gli argomenti replicati dagli altri nel cluster di destinazione ed evitare la replica circolare dei dati tra i cluster.

Ad esempio, MSK Replicator replica i dati in un argomento denominato «topic» dal cluster di origine in un nuovo argomento nel cluster di destinazione denominato **<** alias**>**.topic. sourceKafkaCluster È possibile trovare il prefisso che verrà aggiunto ai nomi degli argomenti nel cluster di destinazione nel campo sourceKafkaClusterAlias utilizzando l'**`DescribeReplicator`**API o la pagina dei dettagli del Replicator sulla console MSK. Il prefisso nel cluster di destinazione è **<** Alias**>**. sourceKafkaCluster

Per garantire che i consumatori possano riavviare in modo affidabile l'elaborazione dal cluster di standby, è necessario configurare i consumatori in modo che leggano i dati degli argomenti utilizzando un operatore wildcard. `.*` Ad esempio, i tuoi consumatori dovrebbero utilizzare. `*topic1`in entrambe le AWS regioni. Questo esempio includerebbe anche un argomento come`footopic1`, quindi regola l'operatore wildcard in base alle tue esigenze.

È necessario utilizzare MSK Replicator che aggiunge un prefisso quando si desidera conservare i dati del replicatore in un argomento separato nel cluster di destinazione, ad esempio per le configurazioni di cluster attivo-attive.

Argomento identico: replica dei nomi

In alternativa all'impostazione predefinita, Amazon MSK Replicator consente di creare un replicatore con la replica degli argomenti impostata su Replica del nome dell'argomento identico (mantieni lo stesso nome degli argomenti nella console). È possibile creare un nuovo Replicator nella AWS regione in cui si trova il cluster MSK di destinazione. Gli argomenti replicati con nomi identici consentono di evitare di riconfigurare i client per la lettura di argomenti replicati.

La replica identica dei nomi degli argomenti (Mantieni lo stesso nome degli argomenti nella console) presenta i seguenti vantaggi:

- Consente di mantenere gli stessi nomi degli argomenti durante il processo di replica, evitando inoltre automaticamente il rischio di cicli di replica infiniti.
- Semplifica la configurazione e il funzionamento di architetture di streaming multicluster, poiché consente di evitare di riconfigurare i client per la lettura degli argomenti replicati.
- Per le architetture cluster attive e passive, la funzionalità di replica dei nomi degli argomenti identici semplifica inoltre il processo di failover, consentendo alle applicazioni di eseguire il failover

senza problemi su un cluster di standby senza richiedere modifiche ai nomi degli argomenti o riconfigurazioni dei client.

- Può essere utilizzata per consolidare più facilmente i dati di più cluster MSK in un unico cluster per l'aggregazione dei dati o l'analisi centralizzata. Ciò richiede la creazione di replicatori separati per ogni cluster di origine e lo stesso cluster di destinazione.
- Può semplificare la migrazione dei dati da un cluster MSK a un altro replicando i dati su argomenti con nomi identici nel cluster di destinazione.

Amazon MSK Replicator utilizza le intestazioni Kafka per evitare automaticamente la replica dei dati sull'argomento da cui provengono, eliminando il rischio di cicli infiniti durante la replica. Un'intestazione è una coppia chiave-valore che può essere inclusa con la chiave, il valore e il timestamp in ogni messaggio Kafka. MSK Replicator incorpora gli identificatori per il cluster di origine e l'argomento nell'intestazione di ogni record replicato. MSK Replicator utilizza le informazioni di intestazione per evitare cicli di replica infiniti. È necessario verificare che i client siano in grado di leggere i dati replicati come previsto.

Tutorial: configurare i cluster di origine e di destinazione per Amazon MSK Replicator

Questo tutorial mostra come configurare un cluster di origine e un cluster di destinazione nella stessa AWS regione o in AWS regioni diverse. Successivamente, puoi utilizzare questi cluster per creare un replicatore Amazon MSK.

Preparare il cluster di origine Amazon MSK

Se disponi già di un cluster di origine MSK creato per il replicatore MSK, assicurati che soddisfi i requisiti descritti in questa sezione. Altrimenti, segui questi passaggi per creare un cluster di origine serverless o assegnato da MSK.

Il processo per la creazione di un cluster di origine del replicatore MSK tra regioni e nella stessa regione è simile. Le differenze vengono evidenziate nelle seguenti procedure.

1. Crea un cluster MSK Serverless o assegnato con il Controllo degli accessi IAM attivato nella regione di origine. Il cluster di origine deve avere un minimo di tre broker.
2. Per un replicatore MSK tra regioni, se l'origine è un cluster assegnato, configuralo con la connettività privata multi-VPC attivata per gli schemi di Controllo degli accessi IAM. Tieni presente che il tipo di autenticazione non autenticato non è supportato quando è attivato il multi-VPC. Non

è necessario attivare la connettività privata multi-VPC per altri schemi di autenticazione (MTL) o schemi di SASL/SCRAM). You can simultaneously use mTLS or SASL/SCRAM autenticazione per gli altri client che si connettono al cluster MSK. È possibile configurare la connettività privata multi-VPC tramite le Impostazioni di rete nei dettagli del cluster della console oppure tramite l'API `UpdateConnectivity`. Consulta la sezione Il proprietario del cluster attiva il multi-VPC. Se il cluster di origine è un cluster MSK Serverlesss, non è necessario attivare la connettività privata multi-VPC.

Per un replicatore MSK nella stessa regione, il cluster di origine MSK non richiede una connettività privata multi-VPC e altri client possono comunque accedere al cluster utilizzando il tipo di autenticazione non autenticato.

3. Per i replicatori MSK tra regioni, è necessario collegare una policy di autorizzazione basata sulle risorse al cluster di origine. Ciò consente a MSK di connettersi a questo cluster per replicare i dati. È possibile eseguire questa operazione utilizzando le procedure CLI o AWS Console riportate di seguito. Consulta anche la sezione Policy basate sulle risorse di Amazon MSK. Non è necessario eseguire questa operazione per i replicatori MSK nella stessa regione.

Console: create resource policy

Aggiorna la policy del cluster di origine con il seguente codice JSON. Sostituisci il segnaposto con l'ARN del tuo cluster di origine.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Service": [
                "kafka.amazonaws.com"
            ]
        },
        "Action": [
            "kafka:CreateVpcConnection",
            "kafka:GetBootstrapBrokers",
            "kafka:DescribeClusterV2"
        ],
```

```
        "Resource": "arn:aws:kafka:us-
east-1:123456789012:cluster/myCluster/abcd1234-5678-90ab-cdef-1234567890ab-1"
    }
  ]
}
```

Utilizza l'opzione Modifica policy del cluster nel menu Operazioni nella pagina dei dettagli del cluster.

CLI: create resource policy

Nota: se utilizzi la AWS console per creare un cluster di origine e scegli l'opzione per creare un nuovo ruolo IAM, AWS allega la policy di fiducia richiesta al ruolo. Se invece desideri che MSK utilizzi un ruolo IAM esistente o se crei un ruolo autonomamente, collega la seguente policy di attendibilità a tale ruolo in modo che il replicatore MSK possa assumerlo. Per informazioni su come modificare la relazione di trust di un ruolo, consulta Modifica di un ruolo.

1. Recupera la versione corrente della policy del cluster MSK utilizzando questo comando. Sostituisci i segnaposto con l'ARN effettivo del cluster.

```
aws kafka get-cluster-policy –cluster-arn <Cluster ARN>
{
"CurrentVersion": "K1PA6795UKM GR7",
"Policy": "..."
}
```

2. Crea una policy basata sulle risorse per consentire al replicatore MSK di accedere al cluster di origine. Utilizza la seguente sintassi come modello sostituendo il segnaposto con l'ARN effettivo del cluster di origine.

```
aws kafka put-cluster-policy --cluster-arn "<sourceClusterARN>" --policy '{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Principal": {
"Service": [
"kafka.amazonaws.com"
]
},
```

```
"Action": [
"kafka:CreateVpcConnection",
"kafka:GetBootstrapBrokers",
"kafka:DescribeClusterV2"
],
"Resource": "<sourceClusterARN>"
}
]
```

Preparare il cluster di destinazione Amazon MSK

Crea un cluster di destinazione MSK (assegnato o serverless) con il Controllo degli accessi IAM attivato. Il cluster di destinazione non richiede l'attivazione della connettività privata multi-VPC. Il cluster di destinazione può trovarsi nella stessa AWS regione o in una regione diversa del cluster di origine. Sia il cluster di origine che quello di destinazione devono trovarsi nello stesso AWS account. Il cluster di destinazione deve avere un minimo di tre broker.

Tutorial: creare un Amazon MSK Replicator

Dopo aver configurato i cluster di origine e di destinazione, puoi utilizzare tali cluster per creare un Amazon MSK Replicator. Prima di creare un replicatore Amazon MSK, assicurati di disporre delle Autorizzazioni IAM necessarie per creare un replicatore MSK.

Argomenti

Considerazioni sulla creazione di un Amazon MSK Replicator

Le seguenti sezioni forniscono una panoramica dei prerequisiti, delle configurazioni supportate e delle best practice per l'utilizzo della funzionalità MSK Replicator. Descrive le autorizzazioni necessarie, la compatibilità dei cluster e i requisiti specifici di Serverless, oltre a indicazioni sulla gestione del Replicator dopo la creazione.

Autorizzazioni IAM necessarie per creare un replicatore MSK

Ecco un esempio della policy IAM necessaria per creare un replicatore MSK. L'operazione `kafka:TagResource` è necessaria solo durante la creazione del replicatore MSK vengono forniti dei tag. Le policy IAM di Replicator devono essere associate al ruolo IAM corrispondente al cliente. Per informazioni sulla creazione di politiche di autorizzazione, consulta Creare politiche di autorizzazione.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MSKReplicatorIAMPassRole",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/MSKReplicationRole",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "kafka.amazonaws.com"
        }
      }
    },
    {
      "Sid": "MSKReplicatorServiceLinkedRole",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::123456789012:role/aws-service-role/
kafka.amazonaws.com/AWSServiceRoleForKafka*"
    },
    {
      "Sid": "MSKReplicatorEC2Actions",
      "Effect": "Allow",
```

```
      "Action": [
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVpcs",
        "ec2:CreateNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:123456789012:subnet/subnet-0abcd1234ef56789",
        "arn:aws:ec2:us-east-1:123456789012:security-group/sg-0123abcd4567ef89",
        "arn:aws:ec2:us-east-1:123456789012:network-
interface/eni-0a1b2c3d4e5f67890",
        "arn:aws:ec2:us-east-1:123456789012:vpc/vpc-0a1b2c3d4e5f67890"
      ]
    },
    {
      "Sid": "MSKReplicatorActions",
      "Effect": "Allow",
      "Action": [
        "kafka:CreateReplicator",
        "kafka:TagResource"
      ],
      "Resource": [
        "arn:aws:kafka:us-
east-1:123456789012:cluster/myCluster/abcd1234-56ef-78gh-90ij-klmnopqrstuv",
        "arn:aws:kafka:us-
east-1:123456789012:replicator/myReplicator/wxyz9876-54vu-32ts-10rq-ponmlkjihgfe"
      ]
    }
  ]
}
```

Di seguito è riportata una policy IAM di esempio per descrivere il replicatore. È necessario specificare l'operazione `kafka:DescribeReplicator` o l'operazione `kafka:ListTagsForResource`, ma non entrambe.

JSON

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

```
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "kafka:DescribeReplicator",
                "kafka:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

Tipi e versioni di cluster supportati per MSK Replicator

Questi sono i requisiti per i tipi di istanze supportati, le versioni di Kafka e le configurazioni di rete.

- Il replicatore MSK supporta sia i cluster assegnati da MSK sia i cluster MSK Serverless in qualsiasi combinazione di cluster di origine e di destinazione. Al momento, il replicatore MSK non supporta altri tipi di cluster Kafka.
- I cluster MSK Serverless richiedono il Controllo degli accessi IAM, non supportano la replica delle ACL di Apache Kafka e offrono un supporto limitato per la replica della configurazione sull'argomento. Consultare Cos'è MSK Serverless?.
- MSK Replicator è supportato solo su cluster che eseguono Apache Kafka 2.7.0 o versioni successive, indipendentemente dal fatto che i cluster di origine e di destinazione si trovino nello stesso o in modo diverso. Regioni AWS
- MSK Replicator supporta i cluster che utilizzano tipi di istanze m5.large o superiori. I cluster t3.small non sono supportati.
- Se si utilizza il replicatore MSK con un cluster assegnato da MSK, sono necessari almeno tre broker sia nel cluster di origine sia in quello di destinazione. È possibile replicare i dati tra cluster in due zone di disponibilità, ma in tali cluster sono necessari almeno quattro broker.
- I cluster MSK di origine e di destinazione devono appartenere allo stesso account. AWS La replica tra cluster in account diversi non è supportata.
- Se i cluster MSK di origine e di destinazione si trovano in AWS regioni diverse (interregionali), MSK Replicator richiede che il cluster di origine abbia la connettività privata multi-VPC attivata per il metodo IAM Access Control.

  Il multi-VPC non è richiesto per altri metodi di autenticazione sul cluster di origine per la replica MSK su tutto il cluster. Regioni AWS

Inoltre, il multi-VPC non è necessario se si replicano dati tra cluster nello stesso. Regione AWS Consultare the section called “Connettività privata multi-VPC in un'unica regione”.

- La replica identica dei nomi degli argomenti (Mantieni lo stesso nome degli argomenti nella console) richiede un cluster MSK con Kafka versione 2.8.1 o successiva.
- Per le configurazioni di replica identica dei nomi degli argomenti (Mantieni lo stesso nome degli argomenti nella console), per evitare il rischio di replica ciclica, non apportare modifiche alle intestazioni create da MSK Replicator (). `__mskmr`

Configurazione del cluster MSK Serverless supportata

- MSK Serverless supporta la replica di queste configurazioni di argomenti per i cluster di destinazione MSK Serverless durante la creazione degli argomenti: `cleanup.policy`, `compression.type`, `max.message.bytes`, `retention.bytes`, `retention.ms`.
- MSK Serverless supporta solo queste configurazioni degli argomenti durante la sincronizzazione della configurazione degli argomenti: `compression.type`, `max.message.bytes`, `retention.bytes`, `retention.ms`.
- Il replicatore utilizza 83 partizioni compattate sui cluster MSK Serverless di destinazione. Assicurati che i cluster MSK Serverless di destinazione abbiano un numero sufficiente di partizioni compattate. Consultare Quota di MSK Serverless.

Modifiche alla configurazione del cluster

- Si consiglia di non attivare o disattivare l'archiviazione a più livelli dopo la creazione del replicatore MSK. Se il cluster di destinazione non è a più livelli, MSK non copierà le configurazioni di archiviazione a più livelli, indipendentemente dal fatto che il cluster di origine sia a più livelli o meno. Se si attiva l'archiviazione a più livelli sul cluster di destinazione dopo la creazione del replicatore, è necessario ricreare il replicatore. Se si desidera copiare i dati da un cluster non a più livelli a un cluster a più livelli, non è necessario copiare le configurazioni degli argomenti. Consulta la sezione Abilitazione e disabilitazione dell'archiviazione a più livelli su un argomento esistente.
- Non modificare le impostazioni di configurazione del cluster dopo la creazione del replicatore MSK. Le impostazioni di configurazione del cluster vengono convalidate durante la creazione del replicatore MSK. Per evitare problemi con il replicatore MSK, non modificare le seguenti impostazioni dopo la creazione dello stesso.
  - Modifica il cluster MSK nel tipo di istanza t3.

- Modifica i permessi del ruolo di esecuzione del servizio.
- Disabilita la connettività privata multi-VPC di MSK.
- Modifica la policy basata sulle risorse del cluster collegata.
- Modifica le regole del gruppo di sicurezza del cluster.

Creazione di un replicatore tramite la console AWS nella regione del cluster di destinazione

La sezione seguente spiega passo per passo il flusso di lavoro della console per la creazione di un replicatore.

Dettagli del replicatore

1. Nella AWS regione in cui si trova il cluster MSK di destinazione, apri la console Amazon MSK da casahttps://console.aws.amazon.com/msk/? region=us-east-1#/home/.
2. Scegli Replicatori per visualizzare l'elenco dei replicatori presenti nell'account.
3. Scegli Crea replicatore.
4. Nel riquadro Dettagli del replicatore, assegna un nome univoco al nuovo replicatore.

Scelta del cluster di origine

Il cluster di origine contiene i dati da copiare in un cluster MSK di destinazione.

1. Nel riquadro Cluster di origine, scegli la regione AWS in cui si trova il cluster di origine.

   È possibile cercare la regione di un cluster accedendo a Cluster MSK e controllando i dettagli dell'ARN in Cluster. Il nome della regione è incorporato nella stringa ARN. Nell'ARN di esempio seguente, il cluster primario si trova nella regione `ap-southeast-2`.

   ```
   arn:aws:kafka:ap-southeast-2:123456789012:cluster/cluster-11/
   eec93c7f-4e8b-4baf-89fb-95de01ee639c-s1
   ```

2. Inserisci l'ARN del tuo cluster di origine o naviga per scegliere il tuo cluster di origine.
3. Scegli una o più sottoreti per il cluster di origine.

   La console mostra le sottoreti disponibili nella regione del cluster di origine che puoi selezionare. È necessario selezionare almeno due sottoreti. Per un replicatore MSK nella stessa regione, le

sottoreti selezionate sono impostate per accedere al cluster di origine e le sottoreti per accedere al cluster di destinazione devono trovarsi nella stessa zona di disponibilità.

4. Scegliete uno o più gruppi di sicurezza per MSK Replicator per accedere al cluster di origine.

   - Per la replica tra regioni (CRR), non è necessario fornire gruppi di sicurezza per il cluster di origine.
   - Per la replica nella stessa regione (SRR), accedi alla EC2 console Amazon all'indirizzo https://console.aws.amazon.com/ec2/ e assicurati che i gruppi di sicurezza che fornirai per il Replicator dispongano di regole in uscita per consentire il traffico verso i gruppi di sicurezza del cluster di origine. Inoltre, assicurati che i gruppi di sicurezza del cluster di origine dispongano di regole in entrata che consentano il traffico proveniente dai gruppi di sicurezza Replicator forniti per l'origine.

     Per aggiungere regole in entrata al gruppo di sicurezza del cluster di origine:

     1. Nella AWS console, accedi ai dettagli del cluster di origine selezionando il nome del cluster.
     2. Seleziona la scheda Proprietà, quindi scorri verso il basso fino al riquadro Impostazioni di rete e seleziona il nome del Gruppo di sicurezza applicato.
     3. Vai alle regole in entrata e seleziona Modifica le regole in entrata.
     4. Seleziona Aggiungi regola.
     5. Nella colonna Tipo per la nuova regola, seleziona TCP personalizzato.
     6. Nella colonna Intervallo di porte, digita`9098`. MSK Replicator utilizza il controllo degli accessi IAM per connettersi al cluster che utilizza la porta 9098.
     7. Nella colonna Origine, digita il nome del gruppo di sicurezza che fornirai durante la creazione di Replicator per il cluster di origine (potrebbe essere lo stesso del gruppo di sicurezza del cluster di origine MSK), quindi seleziona Salva regole.

     Per aggiungere regole in uscita al gruppo di sicurezza di Replicator fornito per l'origine:

     1. Nella AWS console per Amazon EC2, vai al gruppo di sicurezza che fornirai durante la creazione di Replicator per l'origine.
     2. Vai alle regole in uscita e seleziona Modifica regole in uscita.
     3. Seleziona Aggiungi regola.

4. Nella colonna Tipo per la nuova regola, seleziona TCP personalizzato.
5. Nella colonna Intervallo di porte, digita`9098`. MSK Replicator utilizza il controllo degli accessi IAM per connettersi al cluster che utilizza la porta 9098.
6. Nella colonna Origine, digita il nome del gruppo di sicurezza del cluster di origine MSK, quindi seleziona Salva regole.

> **Note**
>
> In alternativa, se non desideri limitare il traffico utilizzando i tuoi gruppi di sicurezza, puoi aggiungere regole in entrata e in uscita che consentono All Traffic.
> 1. Seleziona Aggiungi regola.
> 2. Nella colonna Tipo, seleziona Tutto il traffico.
> 3. Nella colonna Origine, digita `0.0.0.0/0` e quindi seleziona Salva regole.

Scelta del cluster di destinazione

Il cluster di destinazione è il cluster MSK assegnato o serverless in cui vengono copiati i dati di origine.

> **Note**
>
> Il replicatore MSK crea nuovi argomenti nel cluster di destinazione con un prefisso generato automaticamente aggiunto al nome dell'argomento. Ad esempio, il replicatore MSK replica i dati in "`topic`" dal cluster di origine a un nuovo argomento nel cluster di destinazione denominato `<sourceKafkaClusterAlias>.topic`. Questo serve a distinguere gli argomenti che contengono i dati replicati dal cluster di origine da altri argomenti del cluster di destinazione ed evitare che i dati vengano replicati circolarmente tra i cluster. È possibile trovare il prefisso che verrà aggiunto ai nomi degli argomenti nel cluster di destinazione nel campo sourceKafkaClusterAlias utilizzando l'`DescribeReplicator`API o nella pagina dei dettagli del Replicator sulla console MSK. Il prefisso nel cluster di destinazione è. `<sourceKafkaClusterAlias>`

1. Nel riquadro Cluster di destinazione, scegli la AWS regione in cui si trova il cluster di destinazione.

2. Inserisci l'ARN del cluster di destinazione o sfoglia l'elenco per scegliere il cluster di destinazione.
3. Scegli una o più sottoreti per il tuo cluster di destinazione.

   La console mostra le sottoreti disponibili nella regione del cluster di destinazione che puoi selezionare. È necessario selezionare almeno due sottoreti.
4. Scegliete i gruppi di sicurezza per MSK Replicator per accedere al cluster di destinazione.

   Vengono visualizzati i gruppi di sicurezza disponibili nella regione del cluster di destinazione che puoi selezionare. Il gruppo di sicurezza scelto è associato a ciascuna connessione. Per ulteriori informazioni sull'uso dei gruppi di sicurezza, consulta la sezione Controlla il traffico verso AWS le tue risorse utilizzando i gruppi di sicurezza nella Amazon VPC User Guide.

   - Sia per la replica tra regioni (CRR) che per la replica su stessa regione (SRR), accedi alla EC2 console di Amazon all'indirizzo https://console.aws.amazon.com/ec2/e assicurati che i gruppi di sicurezza che fornirai al Replicator dispongano di regole in uscita per consentire il traffico verso i gruppi di sicurezza del cluster di destinazione. Inoltre, assicurati che i gruppi di sicurezza del cluster di destinazione dispongano di regole in entrata che consentano il traffico verso i gruppi di sicurezza del replicatore forniti per la destinazione.

     Per aggiungere regole in entrata al gruppo di sicurezza del cluster di destinazione:

     1. Nella AWS console, accedi ai dettagli del cluster di destinazione selezionando il nome del cluster.
     2. Seleziona la scheda Proprietà, quindi scorri verso il basso fino al riquadro Impostazioni di rete per selezionare il nome del gruppo di sicurezza applicato.
     3. Vai alle regole in entrata e seleziona Modifica le regole in entrata.
     4. Seleziona Aggiungi regola.
     5. Nella colonna Tipo per la nuova regola, seleziona TCP personalizzato.
     6. Nella colonna Intervallo di porte, digita`9098`. MSK Replicator utilizza il controllo degli accessi IAM per connettersi al cluster che utilizza la porta 9098.
     7. Nella colonna Origine, digita il nome del gruppo di sicurezza che fornirai durante la creazione di Replicator per il cluster di destinazione (potrebbe essere lo stesso del gruppo di sicurezza del cluster di destinazione MSK), quindi seleziona Salva regole.

Per aggiungere regole in uscita al gruppo di sicurezza di Replicator fornito per la destinazione:

1. Nella AWS console, vai al gruppo di sicurezza che fornirai durante la creazione di Replicator per la destinazione.
2. Seleziona la scheda Proprietà, quindi scorri verso il basso fino al riquadro Impostazioni di rete per selezionare il nome del gruppo di sicurezza applicato.
3. Vai alle regole in uscita e seleziona Modifica regole in uscita.
4. Seleziona Aggiungi regola.
5. Nella colonna Tipo per la nuova regola, seleziona TCP personalizzato.
6. Nella colonna Intervallo di porte, digita`9098`. MSK Replicator utilizza il controllo degli accessi IAM per connettersi al cluster che utilizza la porta 9098.
7. Nella colonna Origine, digita il nome del gruppo di sicurezza del cluster di destinazione MSK, quindi seleziona Salva regole.

> **Note**
>
> In alternativa, se non desideri limitare il traffico utilizzando i tuoi gruppi di sicurezza, puoi aggiungere regole in entrata e in uscita che consentono All Traffic.
> 1. Seleziona Aggiungi regola.
> 2. Nella colonna Tipo, seleziona Tutto il traffico.
> 3. Nella colonna Origine, digita `0.0.0.0/0` e quindi seleziona Salva regole.

Configurazione delle impostazioni e delle autorizzazioni del replicatore

1. Nel riquadro Impostazioni del replicatore, specifica gli argomenti che desideri replicare utilizzando le espressioni regolari negli elenchi consentiti e non consentiti. Come impostazione predefinita, vengono replicati tutti gli argomenti.

   > **Note**
   >
   > MSK Replicator replica solo fino a 750 argomenti in ordine ordinato. Se è necessario replicare più argomenti, si consiglia di creare un Replicator separato. Vai al Support Center della AWS console e crea un caso di supporto se hai bisogno di supporto per più

di 750 argomenti per Replicator. È possibile monitorare il numero di argomenti replicati utilizzando la metrica TopicCount "». Consultare Quota di broker Amazon MSK Standard.

2. Per impostazione predefinita, MSK Replicator avvia la replica dall'offset più recente (più recente) negli argomenti selezionati. In alternativa, è possibile avviare la replica dal primo offset (il più vecchio) negli argomenti selezionati se si desidera replicare i dati esistenti sugli argomenti. Una volta creato il Replicator, non è possibile modificare questa impostazione. Questa impostazione corrisponde al `startingPosition`campo della `CreateReplicator`richiesta e della `DescribeReplicator`risposta APIs.
3. Scegli la configurazione del nome di un argomento:

   - `PREFIXED`replica del nome dell'argomento (aggiungi il prefisso al nome dell'argomento nella console): l'impostazione predefinita. MSK Replicator replica «topic1" dal cluster di origine in un nuovo argomento nel cluster di destinazione con lo stesso nome. `<sourceKafkaClusterAlias>.topic1`
   - Replica identica dei nomi degli argomenti (mantieni lo stesso nome degli argomenti nella console): gli argomenti del cluster di origine vengono replicati con nomi di argomento identici nel cluster di destinazione.

   Questa impostazione corrisponde al `TopicNameConfiguration` campo nella `CreateReplicator` richiesta e `DescribeReplicator` nella risposta. APIs Consultare Come funziona il replicatore Amazon MSK.

   > **Note**
   >
   > Per impostazione predefinita, MSK Replicator crea nuovi argomenti nel cluster di destinazione con un prefisso generato automaticamente aggiunto al nome dell'argomento. Questo serve a distinguere gli argomenti che contengono i dati replicati dal cluster di origine da altri argomenti del cluster di destinazione ed evitare che i dati vengano replicati circolarmente tra i cluster. In alternativa, è possibile creare un replicatore MSK con replica identica dei nomi degli argomenti (mantenere lo stesso nome degli argomenti nella console) in modo che i nomi degli argomenti vengano conservati durante la replica. Questa configurazione riduce la necessità di riconfigurare le applicazioni client durante la configurazione e semplifica il funzionamento di architetture di streaming multicluster.

4. Per impostazione predefinita, MSK Replicator copia tutti i metadati, incluse le configurazioni degli argomenti, le liste di controllo degli accessi () ACLs e gli offset dei gruppi di consumatori per un failover senza interruzioni. Se non si sta creando il replicatore per il failover, è possibile scegliere facoltativamente di disattivare una o più di queste impostazioni disponibili nella sezione Impostazioni aggiuntive.

   > **Note**
   >
   > MSK Replicator non replica la scrittura ACLs poiché i produttori non dovrebbero scrivere direttamente sull'argomento replicato nel cluster di destinazione. I produttori devono scrivere sull'argomento locale nel cluster di destinazione dopo il failover. Per informazioni dettagliate, vedi Esegui un failover pianificato nella regione secondaria AWS.

5. Nel riquadro Replica del gruppo di consumatori, specifica i gruppi di consumatori che desideri replicare utilizzando le espressioni regolari negli elenchi consentiti e non consentiti. Per impostazione predefinita, vengono replicati tutti i gruppi di consumatori.
6. Nel riquadro Compressione, puoi facoltativamente scegliere di comprimere i dati scritti nel cluster di destinazione. Se intendi utilizzare la compressione, ti consigliamo di utilizzare lo stesso metodo di compressione dei dati nel cluster di origine.
7. Nel riquadro Autorizzazioni di accesso, effettua una delle operazioni seguenti:

   a. Seleziona Crea o aggiorna il ruolo IAM con le policy richieste. La console MSK collegherà automaticamente le autorizzazioni e la policy di attendibilità necessarie al ruolo di esecuzione del servizio richiesto per la lettura e la scrittura nei cluster MSK di origine e di destinazione.

   b. Fornisci il tuo ruolo IAM selezionando Scegli tra i ruoli IAM che Amazon MSK può assumere. Ti consigliamo di collegare la policy IAM `AWSMSKReplicatorExecutionRole` gestita al tuo ruolo di esecuzione del servizio, anziché scrivere la tua politica IAM.

      - Crea il ruolo IAM che il Replicator utilizzerà per leggere e scrivere nei cluster MSK di origine e di destinazione utilizzando il codice JSON riportato di seguito come parte della policy di fiducia e come `AWSMSKReplicatorExecutionRole` allegato al ruolo. Nella policy di attendibilità, sostituisci il segnaposto <yourAccountID> con l'ID dell'account effettivo.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "kafka.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<yourAccountID>"
                }
            }
        }
    ]
}
```

8. Nel riquadro Tag del replicatore, puoi facoltativamente assegnare tag alla risorsa replicatore MSK. Per ulteriori informazioni, consulta Contrassegna un tag a un cluster Amazon MSK. Per un replicatore MSK tra regioni, i tag vengono sincronizzati automaticamente con la regione remota al momento della creazione del replicatore. Se si modificano i tag dopo la creazione del replicatore, la modifica non viene sincronizzata automaticamente con la regione remota, quindi sarà necessario sincronizzare manualmente i riferimenti del replicatore locale e del replicatore remoto.
9. Seleziona Crea.

Se desideri limitare le `kafka-cluster:WriteData` autorizzazioni, consulta la sezione Creare politiche di autorizzazione di Come funziona il controllo degli accessi IAM per Amazon MSK. Dovrai aggiungere l'`kafka-cluster:WriteDataIdempotently`autorizzazione sia al cluster di origine che a quello di destinazione.

La creazione e la transizione del replicatore MSK allo stato RUNNING richiedono circa 30 minuti.

Se si crea un nuovo replicatore MSK per sostituirne uno eliminato, il nuovo replicatore avvia la replica dall'offset più recente.

Se il replicatore MSK è passato allo stato FAILED, consulta la sezione Risoluzione dei problemi relativi al replicatore MSK.

Modifica delle impostazioni del replicatore MSK

Non è possibile modificare il cluster di origine, il cluster di destinazione, la posizione iniziale del Replicator o la configurazione di replica dei nomi degli argomenti una volta creato MSK Replicator. È necessario creare un nuovo replicatore per utilizzare la configurazione di replica dei nomi degli argomenti identici. Tuttavia, è possibile modificare altre impostazioni del Replicator, ad esempio argomenti e gruppi di consumatori, da replicare.

1. Accedere a e aprire la console Amazon MSK da https://console.aws.amazon.com/msk/casa?AWS Management Console region=us-east-1#/home/.
2. Nel riquadro di navigazione a sinistra, scegli Replicatori per visualizzare l'elenco dei replicatori presenti nell'account e seleziona il Replicatore MSK che desideri modificare.
3. Scegliere la scheda Properties (Proprietà).
4. Nella sezione Impostazioni del replicatore, scegli Modifica replicatore.
5. È possibile modificare le impostazioni del replicatore MSK modificando una qualsiasi di queste impostazioni.

   - Specifica gli argomenti che desideri replicare utilizzando le espressioni regolari negli elenchi consentiti e non consentiti. Per impostazione predefinita, MSK Replicator copia tutti i metadati, tra cui le configurazioni degli argomenti, le liste di controllo degli accessi (ACLs) e gli offset dei gruppi di consumatori per un failover senza interruzioni. Se non si sta creando il replicatore per il failover, è possibile scegliere facoltativamente di disattivare una o più di queste impostazioni disponibili nella sezione Impostazioni aggiuntive.

     > **Note**
     >
     > MSK Replicator non replica la scrittura ACLs poiché i produttori non dovrebbero scrivere direttamente sull'argomento replicato nel cluster di destinazione. I produttori devono scrivere sull'argomento locale nel cluster di destinazione dopo il failover. Per informazioni dettagliate, vedi Esegui un failover pianificato nella regione secondaria AWS.

   - In Replica del gruppo di consumatori, puoi specificare i gruppi di consumatori che desideri replicare utilizzando le espressioni regolari negli elenchi consentiti e non consentiti. Per

impostazione predefinita, vengono replicati tutti i gruppi di consumatori. Se gli elenchi consentiti e non consentiti sono vuoti, la replica dei gruppi di consumatori è disattivata.

- In Tipo di compressione di destinazione, puoi scegliere se comprimere i dati scritti nel cluster di destinazione. Se intendi utilizzare la compressione, ti consigliamo di utilizzare lo stesso metodo di compressione dei dati nel cluster di origine.

6. Salvare le modifiche.

   La creazione e la transizione del replicatore MSK allo stato di esecuzione richiedono circa 30 minuti. Se il replicatore MSK è passato allo stato FAILED, consulta la sezione ??? sulla risoluzione dei problemi.

Eliminazione di un replicatore MSK

Se la creazione di un replicatore MSK ha esito negativo (stato FAILED), potrebbe essere necessario eliminarlo. I cluster di origine e di destinazione assegnati a un replicatore MSK non possono essere modificati una volta creato il replicatore MSK. È possibile eliminare un replicatore MSK esistente e crearne uno nuovo. Se si crea un nuovo replicatore MSK per sostituire quello eliminato, il nuovo replicatore avvia la replica dall'offset più recente.

1. Nella AWS regione in cui si trova il cluster di origine, accedi e apri la console Amazon MSK da https://console.aws.amazon.com/msk/casa? AWS Management Console region=us-east-1#/home/.
2. Nel riquadro di navigazione, seleziona Replicatori.
3. Dall'elenco dei replicatori MSK, seleziona quello che desideri eliminare e scegli Elimina.

Monitoraggio della replica

È possibile utilizzarlo https://console.aws.amazon.com/cloudwatch/nella regione del cluster di destinazione per `ReplicationLatency` visualizzare i parametri e `ReplicatorThroughput` a livello di argomento e aggregazione per ogni Amazon MSK Replicator. `MessageLag` Le metriche sono visibili ReplicatorNamenello spazio dei nomi «/Kafka».AWS Per verificare la presenza di problemi, puoi anche consultare i parametri `ReplicatorFailure`, `AuthError` e `ThrottleTime`.

La console MSK visualizza un sottoinsieme di metriche per ogni MSK Replicator. CloudWatch Dall'elenco dei Replicatori della console, seleziona il nome di un replicatore e scegli la scheda Monitoraggio.

Parametri del replicatore MSK

I parametri seguenti descrivono i parametri delle prestazioni o delle connessioni per il replicatore MSK.

AuthError le metriche non coprono gli errori di autenticazione a livello di argomento. Per monitorare gli errori di autenticazione a livello di argomento di MSK Replicator, monitorate le metriche di Replicator e le metriche a livello di argomento del ReplicationLatency cluster di origine,. MessagesInPerSec Se un argomento è ReplicationLatency stato ridotto a 0 ma l'argomento contiene ancora dati in corso, significa che il Replicator ha un problema di autenticazione con l'argomento. Verifica che il ruolo IAM per l'esecuzione del servizio del replicatore disponga di autorizzazioni sufficienti per accedere all'argomento.

Tipo di parametr	Parametro	Descrizione	Dimensi i	Unità	Granular tà dei parametr grezzi	Statistic he di aggrega one dei parametr grezzi	
Prestazic ni	Replicati onLatency	Tempo impiegato dai record per la replica dal cluster di origine a quello di destinazione; tempo che intercorre tra l'ora di produzion e di un record all'origine e l'ora di replica alla destinazione. Se Replicati onLatency	Replicat rName	Milliseco ndi	Partizion e	Massimo	
			Replicat rName, Argomer	Milliseco ndi	Partizion e	Massimo	

Tipo di parametr	Parametro	Descrizione	Dimensi i	Unità	Granular tà dei parametr grezzi	Statistic he di aggrega one dei parametr grezzi	
		aumenta, controlla se i cluster hanno partizioni sufficienti per supportare la replica. Una latenza di replica elevata può verificar si quando il numero di partizioni è troppo basso per una velocità di trasmissione effettiva elevata.					

Tipo di parametr	Parametro	Descrizione	Dimensi i	Unità	Granular tà dei parametr grezzi	Statistic he di aggrega one dei parametr grezzi
Prestazic ni	MessageLag	Monitora la sincroniz zazione tra MSK Replicator e il cluster di origine. MessageLag indica il ritardo tra i messaggi prodotti nel cluster di origine e i messaggi utilizzati dal replicatore. Non è il ritardo tra il cluster di origine e quello di destinazi one. Anche se il cluster di origine non è disponibi le/interrotto, il replicatore finirà di scrivere il messaggio che ha utilizzat o nel cluster di destinazi one. Dopo	Replicato rName	Conteggi	Partizion e	Somma
			Replicato rName, Argomer	Conteggi	Partizion e	Somma

Tipo di parametr	Parametro	Descrizione	Dimensi i	Unità	Granular tà dei parametr grezzi	Statistic he di aggrega one dei parametr grezzi	
		un'interruzione, MessageLa g mostra un aumento che indica il numero di messaggi che il replicatore si trova dietro al cluster di origine e questo può essere monitorat o fino a quando il numero di messaggi non raggiunge 0, a dimostrazione del fatto che il replicatore ha raggiunto il cluster di origine.					

Tipo di parametr	Parametro	Descrizione	Dimensi i	Unità	Granular tà dei parametr grezzi	Statistic he di aggrega one dei parametr grezzi
Prestazic ni	Replicato rBytesInPerSec	Numero medio di byte elaborati dal replicatore al secondo. I dati elaborati da MSK Replicator sono costituiti da tutti i dati ricevuti da MSK Replicato r, inclusi i dati replicati nel cluster di destinazione e i dati filtrati da MSK Replicato r (solo se il Replicator è configurato con la configura zione del nome dell'argomento identico) per evitare che i dati vengano copiati nuovamente sullo stesso argomento da cui hanno avuto	Replicato rName	BytesPe econd	Replicato rName	Somma

Tipo di parametr	Parametro	Descrizione	Dimensio i	Unità	Granular tà dei parametr grezzi	Statistic he di aggrega one dei parametr grezzi	
		origine. Se il Replicator è configurato con la configura zione del nome dell'argomento «con prefisso» , entrambe le `Replicato rBytesInP erSec` `Replicato rThroughp ut` metriche avranno lo stesso valore in quanto nessun dato verrà filtrato da MSK Replicator.					

Tipo di parametr	Parametro	Descrizione	Dimensio i	Unità	Granular tà dei parametr grezzi	Statistic he di aggrega one dei parametr grezzi
Prestazio ni	Replicato rThroughput	Numero medio di byte replicati al secondo. Se si Replicato rThroughput tratta di un argomento, di un controllo KafkaClus terPingSu ccessCount e di AuthError parametri per garantire che il Replicator sia in grado di comunicare con i cluster, controlla te i parametri del cluster per assicurarvi che il cluster non sia inattivo.	Replicato rName	BytesPer econd	Partizion e	Somma
			Replicato rName, Argomer	BytesPer econd	Partizion e	Somma

Tipo di parametr	Parametro	Descrizione	Dimensi i	Unità	Granular tà dei parametr grezzi	Statistic he di aggregaz one dei parametr grezzi	
Esegui il debug	AuthError	Il numero di connessioni con autenticazione non riuscita al secondo. Se questo parametro è superiore a 0, puoi verificar e se la policy del ruolo di esecuzione del servizio per il replicato re è valida e assicurarti che non siano impostate autorizzazioni di rifiuto per le autorizza zioni del cluster. In base alla dimensione clusterAlias, è possibile verificar e se è il cluster di origine o di	Replicato rName, ClusterA ias	Conteggi	Worker	Somma	

Tipo di parametr	Parametro	Descrizione	Dimensio i	Unità	Granular tà dei parametr grezzi	Statistic he di aggrega one dei parametr grezzi	
		destinazione a presentare errori di autentica zione.					

Tipo di parametr	Parametro	Descrizione	Dimensi i	Unità	Granular tà dei parametr grezzi	Statistic he di aggrega one dei parametr grezzi
Esegui il debug	ThrottleTime	Il tempo medio, espresso in millisecondi, per il quale i broker del cluster hanno limitato la larghezza di banda della rete per una richiesta . Imposta la limitazione della larghezza di banda della rete per evitare che il replicatore MSK sovraccar ichi il cluster. Se questo parametro è 0, replicati onLatency non è elevato e replicato rThroughp ut è come previsto, allora la limitazione della larghezza	Replicato rName, ClusterA ias	Milliseco ndi	Worker	Massimo

Tipo di parametr	Parametro	Descrizione	Dimensio i	Unità	Granular tà dei parametr grezzi	Statistic he di aggregaz one dei parametr grezzi	
		di banda della rete funziona come previsto. Se questo parametro è superiore a 0, è possibile regolare la limitazione della larghezza di banda della rete di conseguenza.					
Esegui il debug	ReplicatorFailure	Numero di errori riscontrati dal replicatore.	Replicato rName	Conteggi		Somma	

Tipo di parametr	Parametro	Descrizione	Dimensi i	Unità	Granular tà dei parametr grezzi	Statistic he di aggrega one dei parametr grezzi
Esegui il debug	KafkaClus terPingSu ccessCount	Indica lo stato della connessio ne del replicatore al cluster Kafka. Se questo valore è 1, la connessio ne è integra. Se il valore è 0 o nessun punto di dati, la connessione non è integra. Se il valore è 0, puoi controlla re le impostazi oni di rete o di autorizzazione IAM per il cluster Kafka. In base alla ClusterAlias dimensione, è possibile identific are se questa metrica è per il cluster di origine o di destinazi one.	Replicato rName, ClusterA ias	Conteggi		Somma

Utilizza la replica per aumentare la resilienza di un'applicazione di streaming Kafka in tutte le regioni

È possibile utilizzare MSK Replicator per configurare topologie di cluster attive-attive o attive-passive per aumentare la resilienza dell'applicazione Apache Kafka in tutte le regioni. AWS In una configurazione attiva-attiva, entrambi i cluster MSK eseguono attivamente operazioni di lettura e scrittura. In una configurazione attiva-passiva, solo un cluster MSK alla volta serve attivamente lo streaming di dati, mentre l'altro cluster è in standby.

Considerazioni sulla creazione di applicazioni Apache Kafka multiregionali

I consumatori devono essere in grado di rielaborare i messaggi duplicati senza ripercussioni a valle. MSK Replicator replica i dati at-least-once che possono causare duplicati nel cluster di standby. Quando si passa alla AWS regione secondaria, i consumatori possono elaborare gli stessi dati più di una volta. Il replicatore MSK dà la priorità alla copia dei dati rispetto agli offset dei consumatori per prestazioni migliori. Dopo un failover, il consumatore può iniziare a leggere dagli offset precedenti con conseguente duplicazione dell'elaborazione.

I produttori e i consumatori devono inoltre tollerare una perdita minima di dati. Poiché MSK Replicator replica i dati in modo asincrono, quando la AWS regione primaria inizia a riscontrare errori, non vi è alcuna garanzia che tutti i dati vengano replicati nella regione secondaria. È possibile utilizzare la latenza di replica per determinare il numero massimo di dati che non sono stati copiati nella regione secondaria.

Utilizzo della topologia di cluster attiva-attiva rispetto a quella attiva-passiva

Una topologia di cluster attiva-attiva offre tempi di ripristino quasi nulli e la possibilità per l'applicazione di streaming di funzionare contemporaneamente in più regioni AWS . Quando un cluster in una regione è danneggiato, le applicazioni connesse al cluster nell'altra regione continuano a elaborare i dati.

Le configurazioni attive-passive sono adatte alle applicazioni che possono essere eseguite in una sola regione AWS alla volta o quando è necessario un maggiore controllo sull'ordine di elaborazione dei dati. Le configurazioni attive-passive richiedono più tempo di ripristino rispetto alle configurazioni attive-attive, poiché, dopo un failover, è necessario avviare l'intera configurazione attiva-passiva, compresi produttori e consumatori, nella regione secondaria per riprendere lo streaming dei dati.

Crea una configurazione di cluster Kafka attivo-passiva con le configurazioni di denominazione degli argomenti consigliate

Per una configurazione attiva-passiva, ti consigliamo di utilizzare una configurazione simile di produttori, cluster MSK e consumatori (con lo stesso nome di gruppo di consumatori) in due regioni diverse. AWS È importante che i due cluster MSK abbiano capacità di lettura e scrittura identiche per garantire una replica affidabile dei dati. È necessario creare un replicatore MSK per copiare continuamente i dati dal cluster primario a quello di standby. È inoltre necessario configurare i produttori per scrivere i dati in argomenti su un cluster nella stessa regione. AWS

Per una configurazione attiva-passiva, create un nuovo replicatore con replica identica dei nomi degli argomenti (mantieni lo stesso nome degli argomenti nella console) per iniziare a replicare i dati dal tuo cluster MSK nella regione primaria al cluster nella regione secondaria. Si consiglia di utilizzare un set duplicato di produttori e consumatori nelle due AWS regioni, ciascuno dei quali si connette al cluster della propria regione utilizzando la propria stringa di bootstrap. Ciò semplifica il processo di failover poiché non richiede modifiche alla stringa di bootstrap. Per garantire che i consumatori leggano da dove l'avevano interrotto, i consumatori dei cluster di origine e di destinazione devono avere lo stesso ID del gruppo di consumatori.

Se si utilizza la replica identica dei nomi degli argomenti (Mantieni lo stesso nome degli argomenti nella console) per MSK Replicator, gli argomenti verranno replicati con lo stesso nome degli argomenti di origine corrispondenti.

Si consiglia di configurare le impostazioni e le autorizzazioni a livello di cluster per i client nel cluster di destinazione. Non è necessario configurare le impostazioni a livello di argomento e la lettura letterale ACLs poiché MSK Replicator le copia automaticamente se è stata selezionata l'opzione per copiare le liste di controllo degli accessi. Consultare Replica dei metadati.

Failover nella regione secondaria AWS

Ti consigliamo di monitorare la latenza di replica nella AWS regione secondaria utilizzando Amazon. CloudWatch Durante un evento di servizio nella AWS regione principale, la latenza di replica può aumentare improvvisamente. Se la latenza continua ad aumentare, utilizza il AWS Service Health Dashboard per verificare la presenza di eventi di servizio nella AWS regione principale. Se si verifica un evento, puoi eseguire il failover nella regione secondaria AWS .

Esegui un failover pianificato nella regione secondaria AWS

È possibile eseguire un failover pianificato per testare la resilienza dell'applicazione rispetto a un evento imprevisto nella AWS regione principale in cui si trova il cluster MSK di origine. Un failover pianificato non dovrebbe comportare la perdita di dati.

Se utilizzi un argomento identico alla configurazione della replica dei nomi, segui questi passaggi:

1. Arresta tutti i produttori e i consumatori che si connettono al cluster di origine.
2. Crea un nuovo replicatore MSK per replicare i dati dal cluster MSK nella regione secondaria al cluster MSK nella regione principale con la replica identica dei nomi degli argomenti (mantieni lo stesso nome degli argomenti nella console). Ciò è necessario per copiare i dati che verranno scritti dalla regione secondaria alla regione primaria, in modo da poter eseguire il failback nella regione primaria al termine dell'evento imprevisto.
3. Avvia produttori e consumatori collegati al cluster di destinazione nella regione secondaria. AWS

Se utilizzi la configurazione dei nomi degli argomenti con prefisso, segui questi passaggi per il failover:

1. Arresta tutti i produttori e i consumatori che si connettono al cluster di origine.
2. Crea un nuovo replicatore MSK per replicare i dati dal cluster MSK nella regione secondaria al cluster MSK nella regione primaria. Ciò è necessario per copiare i dati che verranno scritti dalla regione secondaria alla regione primaria, in modo da poter eseguire il failback nella regione primaria al termine dell'evento imprevisto.
3. Avvia i produttori sul cluster target nella regione secondaria AWS .
4. A seconda dei requisiti di ordinamento dei messaggi dell'applicazione, segui i passaggi indicati in una delle schede seguenti.

   No message ordering

   Se l'applicazione non richiede l'ordinamento dei messaggi, avvia i consumatori della AWS regione secondaria che leggono sia gli argomenti locali (ad esempio, topic) che quelli replicati (ad esempio`<sourceKafkaClusterAlias>.topic`) utilizzando un operatore wildcard (ad esempio,). `.*topic`

Message ordering

Se l'applicazione richiede l'ordinamento dei messaggi, avvia i consumatori solo per gli argomenti replicati sul cluster di destinazione (ad esempio, `<sourceKafkaClusterAlias>.topic`) ma non per gli argomenti locali (ad esempio, `topic`).

5. Attendi che tutti i consumatori degli argomenti replicati sul cluster MSK di destinazione completino l'elaborazione di tutti i dati, in modo che il ritardo del consumatore sia 0 e anche il numero di record elaborati sia 0. Quindi, interrompi i consumatori per gli argomenti replicati sul cluster di destinazione. A questo punto, tutti i record replicati dal cluster MSK di origine al cluster MSK di destinazione sono stati utilizzati.
6. Avvia i consumatori per gli argomenti locali (ad esempio, `topic`) sul cluster MSK di destinazione.

Eseguite un failover non pianificato nella regione secondaria AWS

È possibile eseguire un failover non pianificato quando si verifica un evento di servizio nella AWS regione principale in cui è presente il cluster MSK di origine e si desidera reindirizzare temporaneamente il traffico verso la regione secondaria che include il cluster MSK di destinazione. Un failover non pianificato potrebbe causare una perdita di dati poiché MSK Replicator replica i dati in modo asincrono. È possibile tenere traccia del ritardo dei messaggi utilizzando le metriche in. ???

Se utilizzi la configurazione di replica dei nomi degli argomenti identica (mantieni lo stesso nome degli argomenti nella console), segui questi passaggi:

1. Prova a chiudere tutti i produttori e i consumatori che si connettono al cluster MSK di origine nella regione primaria. Questa operazione potrebbe non riuscire a causa di problemi in quella regione.
2. Avvia produttori e consumatori a connettersi al cluster MSK di destinazione nella AWS regione secondaria per completare il failover. Poiché MSK Replicator replica anche i metadati, compresi gli offset di lettura ACLs e gli offset dei gruppi di consumatori, i produttori e i consumatori riprenderanno senza problemi l'elaborazione quasi da dove l'avevano interrotta prima del failover.

Se utilizzi la configurazione del nome dell'`PREFIX`argomento, segui questi passaggi per il failover:

1. Prova a chiudere tutti i produttori e i consumatori che si connettono al cluster MSK di origine nella regione primaria. Questa operazione potrebbe non riuscire a causa di problemi in quella regione.
2. Avvia produttori e consumatori a connettersi al cluster MSK di destinazione nella AWS regione secondaria per completare il failover. Poiché MSK Replicator replica anche i metadati, compresi gli offset di lettura ACLs e gli offset dei gruppi di consumatori, i produttori e i consumatori riprenderanno senza problemi l'elaborazione quasi da dove l'avevano interrotta prima del failover.
3. A seconda dei requisiti di ordinamento dei messaggi dell'applicazione, segui i passaggi indicati in una delle schede seguenti.

   No message ordering

   Se l'applicazione non richiede l'ordinamento dei messaggi, avviate i consumatori della AWS regione di destinazione che leggono sia gli argomenti locali (ad esempio) che quelli replicati (ad esempio`topic`) utilizzando un operatore wildcard (ad esempio,`<sourceKafkaClusterAlias>.topic`). `.*topic`

   Message ordering

   1. Avvia i consumatori solo per gli argomenti replicati nel cluster di destinazione (ad esempio, `<sourceKafkaClusterAlias>.topic`) ma non per gli argomenti locali (ad esempio, `topic`).
   2. Attendi che tutti i consumatori degli argomenti replicati sul cluster MSK di destinazione completino l'elaborazione di tutti i dati, in modo che il ritardo di offset sia 0 e anche il numero di record elaborati sia 0. Quindi, interrompi i consumatori per gli argomenti replicati sul cluster di destinazione. A questo punto, tutti i record replicati dal cluster MSK di origine al cluster MSK di destinazione sono stati utilizzati.
   3. Avvia i consumatori per gli argomenti locali (ad esempio, `topic`) sul cluster MSK di destinazione.
4. Una volta terminato l'evento di servizio nella regione primaria, create un nuovo replicatore MSK per replicare i dati dal cluster MSK nella regione secondaria al cluster MSK nella regione primaria con la posizione iniziale del replicatore impostata sulla prima. Ciò è necessario per copiare i dati che verranno scritti dalla regione secondaria alla regione primaria, in modo da poter eseguire il failback nella regione primaria al termine dell'evento di servizio. Se non impostate la posizione iniziale del Replicator sulla prima, i dati prodotti nel cluster nella regione secondaria durante

l'evento di servizio nell'area primaria non verranno copiati nuovamente nel cluster nell'area primaria.

Eseguire il failback nella regione principale AWS

È possibile eseguire il failback AWS nella regione principale al termine dell'evento di servizio in quella regione.

Se utilizzi la configurazione della replica dei nomi per argomento identico, segui questi passaggi:

1. Crea un nuovo Replicatore MSK con il cluster secondario come origine e il cluster primario come destinazione, impostando la posizione iniziale sulla replica del nome dell'argomento più antica e identica (mantieni lo stesso nome degli argomenti nella console).

   Ciò avvierà il processo di copia di tutti i dati scritti nel cluster secondario dopo il failover nella regione primaria.
2. Monitora la `MessageLag` metrica sul nuovo replicatore in Amazon CloudWatch fino al raggiungimento`0`, il che indica che tutti i dati sono stati replicati dal secondario al primario.
3. Dopo che tutti i dati sono stati replicati, interrompi la connessione di tutti i produttori al cluster secondario e avvia i produttori a connettersi al cluster primario.
4. Attendi `MaxOffsetLag` che i tuoi consumatori si connettano al cluster secondario `0` per assicurarti di aver elaborato tutti i dati. Consultare Monitora i ritardi dei consumatori.
5. Una volta che tutti i dati sono stati elaborati, interrompi i consumatori nell'area secondaria e avvia i consumatori a connettersi al cluster primario per completare il failback.
6. Eliminate il replicatore creato nel primo passaggio, che consiste nella replica dei dati dal cluster secondario a quello primario.
7. Verifica che il Replicator esistente che copia i dati dal cluster primario a quello secondario abbia lo stato «RUNNING» e il `ReplicatorThroughput` parametro in Amazon. CloudWatch `0`

   Tieni presente che quando crei un nuovo Replicator con la posizione iniziale Earliest for failback, inizia a leggere tutti i dati negli argomenti relativi ai cluster secondari. A seconda delle impostazioni di conservazione dei dati, gli argomenti possono contenere dati provenienti dal cluster di origine. Sebbene MSK Replicator filtri automaticamente tali messaggi, dovrete comunque sostenere costi di elaborazione e trasferimento dei dati per tutti i dati del cluster secondario. È possibile tenere traccia del totale dei dati elaborati dal replicatore utilizzando. `ReplicatorBytesInPerSec` Consultare Parametri del replicatore MSK.

Se utilizzi la configurazione dei nomi degli argomenti con prefisso, segui questi passaggi:

È necessario avviare le fasi di failback solo dopo che la replica dal cluster nella regione secondaria al cluster nella regione primaria è stata ripristinata e il MessageLag parametro in CloudWatch Amazon è vicino a 0. Un failback pianificato non dovrebbe comportare la perdita di dati.

1. Arresta tutti i produttori e i consumatori che si connettono al cluster MSK nella regione secondaria.
2. Per la topologia attiva-passiva, elimina il replicatore che replica i dati dal cluster nella regione secondaria alla regione primaria. Non è necessario eliminare il replicatore per la topologia attiva-attiva.
3. Avvia i produttori che si connettono al cluster MSK nella regione primaria.
4. A seconda dei requisiti di ordinamento dei messaggi dell'applicazione, segui i passaggi indicati in una delle schede seguenti.

   No message ordering

   Se la tua applicazione non richiede l'ordinamento dei messaggi, avvia i consumatori della AWS regione primaria che leggono sia gli argomenti locali (ad esempio`topic`) che quelli replicati (ad esempio) utilizzando un operatore wildcard (ad esempio,`<sourceKafkaClusterAlias>.topic`). `.*topic` I consumatori sugli argomenti locali (ad esempio: topic) riprenderanno dall'ultimo offset utilizzato prima del failover. Se erano presenti dati non elaborati prima del failover, verranno elaborati ora. Nel caso di un failover pianificato, tale record non dovrebbe esistere.

   Message ordering

   1. Avvia i consumatori solo per gli argomenti replicati nella regione primaria (ad esempio, `<sourceKafkaClusterAlias>.topic`) ma non per gli argomenti locali (ad esempio, `topic`).
   2. Attendi che tutti i consumatori degli argomenti replicati sul cluster nella regione primaria completino l'elaborazione di tutti i dati, in modo che il ritardo di offset sia 0 e anche il numero di record elaborati sia 0. Quindi, interrompi i consumatori per gli argomenti replicati sul cluster nella regione primaria. A questo punto, tutti i record prodotti nella regione secondaria dopo il failover sono stati utilizzati nella regione primaria.
   3. Avvia i consumatori per gli argomenti locali (ad esempio, `topic`) sul cluster nella regione primaria.

5. Verificate che il replicatore esistente dal cluster nella regione primaria al cluster nella regione secondaria sia nello stato RUNNING e funzioni come previsto utilizzando le `ReplicatorThroughput` metriche di latenza e.

Creare una configurazione attiva-attiva utilizzando MSK Replicator

Se si desidera creare una configurazione attiva-attiva in cui entrambi i cluster MSK eseguano attivamente operazioni di lettura e scrittura, si consiglia di utilizzare un replicatore MSK con replica dei nomi degli argomenti con prefisso (aggiungi il prefisso al nome degli argomenti nella console). Tuttavia, ciò richiederà la riconfigurazione dei consumatori per la lettura degli argomenti replicati.

Segui questi passaggi per configurare la topologia attiva-attiva tra il cluster MSK di origine A e il cluster MSK di destinazione B.

1. Crea un replicatore MSK con il cluster MSK A come origine e il cluster MSK B come destinazione.
2. Dopo aver creato correttamente il replicatore MSK precedente, crea un replicatore con il cluster B come origine e il cluster A come destinazione.
3. Crea due set di produttori, ognuno dei quali scrive i dati contemporaneamente nell'argomento locale (ad esempio, "topic") nel cluster nella stessa regione del produttore.
4. Crea due set di consumatori, ciascuno dei quali legge i dati utilizzando un abbonamento wildcard (ad esempio». *topic») dal cluster MSK nella stessa AWS regione del consumatore. In questo modo, i consumatori leggeranno automaticamente i dati prodotti localmente nella regione dall'argomento locale (ad esempio, `topic`), nonché i dati replicati dall'altra regione nell'argomento con il prefisso `<sourceKafkaClusterAlias>.topic`. Questi due gruppi di consumatori devono avere gruppi di consumatori diversi in IDs modo che gli offset dei gruppi di consumatori non vengano sovrascritti quando MSK Replicator li copia nell'altro cluster.

Se si desidera evitare la riconfigurazione dei client, anziché utilizzare la replica dei nomi degli argomenti con prefisso (aggiungere il prefisso al nome degli argomenti nella console), è possibile creare i replicatori MSK utilizzando la replica dei nomi degli argomenti identici (Mantieni lo stesso nome degli argomenti nella console) per creare una configurazione attiva-attiva. Tuttavia, pagherete costi aggiuntivi per l'elaborazione e il trasferimento dei dati per ogni Replicator. Questo perché ogni replicatore dovrà elaborare il doppio della normale quantità di dati, una volta per la replica e un'altra per evitare cicli infiniti. È possibile tenere traccia della quantità totale di dati elaborati da ciascun replicatore utilizzando la metrica. `ReplicatorBytesInPerSec` Consultare Monitoraggio

della replica. Questa metrica include i dati replicati nel cluster di destinazione e i dati filtrati da MSK Replicator per evitare che i dati vengano ricondotti allo stesso argomento da cui provengono.

> **Note**
>
> Se utilizzi la replica identica dei nomi degli argomenti (Mantieni lo stesso nome degli argomenti nella console) per configurare una topologia attiva-attiva, attendi almeno 30 secondi dopo aver eliminato un argomento prima di ricrearne uno con lo stesso nome. Questo periodo di attesa consente di evitare che i messaggi duplicati vengano replicati nel cluster di origine. I consumatori devono essere in grado di rielaborare i messaggi duplicati senza ripercussioni a valle. Consultare Considerazioni sulla creazione di applicazioni Apache Kafka multiregionali.

Esegui la migrazione da un cluster Amazon MSK a un altro utilizzando MSK Replicator

Puoi utilizzare la replica dei nomi degli argomenti identici per la migrazione dei cluster, ma i tuoi consumatori devono essere in grado di gestire messaggi duplicati senza alcun impatto a valle. Questo perché MSK Replicator fornisce la replica, che può portare a messaggi at-least-once duplicati in rari scenari. Se i tuoi consumatori soddisfano questo requisito, segui questi passaggi.

1. Crea un replicatore che replica i dati dal tuo vecchio cluster al nuovo cluster con la posizione iniziale di Replicator impostata su Earliest e utilizzando la replica identica dei nomi degli argomenti (mantieni lo stesso nome degli argomenti nella console).
2. Configura le impostazioni e le autorizzazioni a livello di cluster sul nuovo cluster. Non è necessario configurare le impostazioni a livello di argomento e la lettura «letterale» ACLs, poiché MSK Replicator le copia automaticamente.
3. Monitora la `MessageLag` metrica in Amazon CloudWatch fino a raggiungere lo 0, il che indica che tutti i dati sono stati replicati.
4. Dopo che tutti i dati sono stati replicati, impedisci ai produttori di scrivere dati sul vecchio cluster.
5. Riconfigura quei produttori per connetterli al nuovo cluster e avviali.
6. Monitora i `MaxOffsetLag` parametri relativi ai consumatori che leggono i dati dal vecchio cluster fino a quando non diventano effettivi`0`, il che indica che tutti i dati esistenti sono stati elaborati.
7. Impedisci ai consumatori che si connettono al vecchio cluster.

8. Riconfigura i consumatori per connettersi al nuovo cluster e avviarli.

Migrazione da MirrorMaker 2 a gestione automatica a MSK Replicator

Per migrare da MirrorMaker (MM2) a MSK Replicator, attenersi alla seguente procedura:

1. Interrompi il produttore che sta scrivendo sul tuo cluster Amazon MSK di origine.
2. Consenti MM2 di replicare tutti i messaggi sugli argomenti dei cluster di origine. È possibile monitorare il ritardo dei MM2 consumatori per ciascun utente sul cluster MSK di origine per determinare quando tutti i dati sono stati replicati.
3. Create un nuovo Replicator con la posizione iniziale impostata su Più recente e la configurazione del nome dell'argomento impostata su `IDENTICAL` (Replica dei nomi degli stessi argomenti nella console).
4. Una volta che il Replicator è nello stato RUNNING, è possibile riavviare i produttori a scrivere nel cluster di origine.

Risolvete i problemi relativi a MSK Replicator

Le seguenti informazioni agevolano la risoluzione dei problemi che si potrebbero verificare con il replicatore MSK. Risolvi i problemi del tuo cluster Amazon MSKPer informazioni sulla risoluzione dei problemi relative ad altre funzionalità di Amazon MSK, consulta. Puoi anche pubblicare il problema in AWS re:Post.

Lo stato del replicatore MSK passa da CREATING a FAILED

Di seguito sono riportate alcune cause comuni degli errori di creazione del replicatore MSK.

1. Assicurati che i gruppi di sicurezza che hai fornito per la creazione del replicatore nella sezione del cluster di destinazione dispongano di regole in uscita per consentire il traffico verso i gruppi di sicurezza del cluster di destinazione. Inoltre, assicurati che i gruppi di sicurezza del cluster di destinazione dispongano di regole in entrata che consentano il traffico verso i gruppi di sicurezza che fornisci per la creazione del replicatore nella sezione del cluster di destinazione. Consultare Scelta del cluster di destinazione.
2. Se stai creando il replicatore per la replica tra regioni, verifica che per il cluster di origine sia attivata la connettività multi-VPC per il metodo di autenticazione Controllo degli accessi IAM.

Consultare Connettività privata multi-VPC di Amazon MSK in un'unica regione. Verifica inoltre che la policy del cluster sia configurata sul cluster di origine in modo che il replicatore MSK possa connettersi a esso. Consultare Preparare il cluster di origine Amazon MSK.

3. Assicurati che il ruolo IAM fornito durante la creazione del replicatore MSK disponga delle autorizzazioni necessarie per leggere e scrivere nei cluster di origine e di destinazione. Inoltre, verifica che il ruolo IAM disponga delle autorizzazioni per scrivere sugli argomenti. Per informazioni, consultare Configurazione delle impostazioni e delle autorizzazioni del replicatore.
4. Verifica che la tua rete non ACLs stia bloccando la connessione tra MSK Replicator e i cluster di origine e di destinazione.
5. È possibile che i cluster di origine o di destinazione non fossero completamente disponibili quando il replicatore MSK ha tentato di connettersi a essi. Ciò potrebbe essere dovuto a un carico eccessivo, all'utilizzo del disco o della CPU, che impedisce al replicatore di connettersi ai broker. Risolvi il problema con i broker e prova di nuovo a creare il replicatore.

Dopo aver eseguito le convalide precedenti, crea nuovamente il replicatore MSK.

Il replicatore MSK appare bloccato nello stato CREATING

A volte la creazione del replicatore MSK può richiedere fino a 30 minuti. Attendi 30 minuti e controlla nuovamente lo stato del replicatore.

Il replicatore MSK non replica dati o replica soltanto dati parziali

Seguire questi passaggi per risolvere i problemi di replica dei dati.

1. Verifica che il tuo Replicator non stia riscontrando errori di autenticazione utilizzando la AuthError metrica fornita da MSK Replicator in Amazon. CloudWatch Se questo parametro è superiore a 0, verifica se la policy del ruolo IAM fornito per il replicatore è valida e che non siano impostate autorizzazioni di rifiuto per le autorizzazioni del cluster. In base alla dimensione clusterAlias, è possibile verificare se è il cluster di origine o quello di destinazione a presentare errori di autenticazione.
2. Verifica che i cluster di origine e di destinazione non presentino problemi. È possibile che il replicatore non sia in grado di connettersi al cluster di origine o di destinazione. Ciò può accadere a causa di un numero eccessivo di connessioni, di un disco a piena capacità o di un elevato utilizzo della CPU.
3. Verifica che i cluster di origine e di destinazione siano raggiungibili da MSK Replicator utilizzando la metrica KafkaClusterPingSuccessCount in Amazon. CloudWatch In base alla dimensione

clusterAlias, è possibile verificare se è il cluster di origine o di destinazione a presentare errori di autenticazione. Se questo parametro è 0 o non ha un punto di dati, la connessione non è integra. È necessario verificare le autorizzazioni di rete e i ruoli IAM utilizzati dal replicatore MSK per connettersi ai cluster.

4. Verifica che il tuo Replicator non stia riscontrando errori dovuti alla mancanza di autorizzazioni a livello di argomento utilizzando la metrica in Amazon. ReplicatorFailure CloudWatch Se questo parametro è superiore a 0, controlla il ruolo IAM che hai fornito per le autorizzazioni a livello di argomento.
5. Verifica che l'espressione regolare che hai fornito nell'elenco consentito durante la creazione del replicatore corrisponda ai nomi degli argomenti che desideri replicare. Inoltre, verifica che gli argomenti non vengano esclusi dalla replica a causa di un'espressione regolare nell'elenco degli argomenti non consentiti.
6. Tieni presente che il Replicator potrebbe impiegare fino a 30 secondi per rilevare e creare nuovi argomenti o partizioni di argomenti sul cluster di destinazione. Tutti i messaggi inviati all'argomento di origine prima della creazione dell'argomento nel cluster di destinazione non verranno replicati se la posizione iniziale del replicatore è la più recente (impostazione predefinita). In alternativa, è possibile avviare la replica dal primo offset nelle partizioni degli argomenti del cluster di origine se si desidera replicare i messaggi esistenti sui propri argomenti nel cluster di destinazione. Consultare Configurazione delle impostazioni e delle autorizzazioni del replicatore.

Gli offset dei messaggi nel cluster di destinazione sono diversi da quelli del cluster di origine

Nell'ambito della replica dei dati, MSK Replicator utilizza i messaggi dal cluster di origine e li produce nel cluster di destinazione. Ciò può portare a messaggi con offset diversi sui cluster di origine e di destinazione. Tuttavia, se è stata attivata la sincronizzazione degli offset dei gruppi di consumatori durante la creazione di Replicator, MSK Replicator tradurrà automaticamente gli offset durante la copia dei metadati in modo che, dopo il failover sul cluster di destinazione, gli utenti possano riprendere l'elaborazione da dove l'avevano interrotta nel cluster di origine.

MSK Replicator non sincronizza gli offset dei gruppi di consumatori oppure il gruppo di consumatori non esiste nel cluster di destinazione

Segui questi passaggi per risolvere i problemi di replica dei metadati.

1. Verifica che la replica dei dati funzioni come previsto. In caso contrario, vediIl replicatore MSK non replica dati o replica soltanto dati parziali.
2. Verifica che l'espressione regolare che hai fornito nell'elenco consentito durante la creazione del Replicator corrisponda ai nomi dei gruppi di consumatori che desideri replicare. Inoltre, verificate che i gruppi di consumatori non vengano esclusi dalla replica a causa di un'espressione regolare nell'elenco degli utenti non autorizzati.
3. Verificate che MSK Replicator abbia creato l'argomento sul cluster di destinazione. Potrebbero essere necessari fino a 30 secondi affinché il Replicator rilevi e crei i nuovi argomenti o le partizioni degli argomenti sul cluster di destinazione. Tutti i messaggi inviati all'argomento di origine prima della creazione dell'argomento nel cluster di destinazione non verranno replicati se la posizione iniziale del replicatore è la più recente (impostazione predefinita). Se il gruppo di consumatori nel cluster di origine ha utilizzato solo i messaggi che non sono stati replicati da MSK Replicator, il gruppo di consumatori non verrà replicato nel cluster di destinazione. Dopo aver creato correttamente l'argomento sul cluster di destinazione, MSK Replicator inizierà a replicare i nuovi messaggi scritti dal cluster di origine alla destinazione. Una volta che il gruppo di consumatori inizia a leggere questi messaggi dall'origine, MSK Replicator replicherà automaticamente il gruppo di consumatori nel cluster di destinazione. In alternativa, è possibile avviare la replica dal primo offset nelle partizioni degli argomenti del cluster di origine se si desidera replicare i messaggi esistenti sui propri argomenti nel cluster di destinazione. Consultare Configurazione delle impostazioni e delle autorizzazioni del replicatore.

> **Note**
>
> MSK Replicator ottimizza la sincronizzazione dell'offset dei gruppi di consumatori per i consumatori del cluster di origine che leggono da una posizione più vicina alla fine della partizione degli argomenti. Se i gruppi di consumatori sono in ritardo rispetto al cluster di origine, è possibile riscontrare un ritardo maggiore per tali gruppi di consumatori sul cluster di destinazione rispetto a quello di origine. Ciò significa che, dopo il failover sul cluster di destinazione, i consumatori rielaboreranno più messaggi duplicati. Per ridurre questo ritardo, i tuoi utenti del cluster di origine dovrebbero recuperare il ritardo e iniziare a consumare dall'estremità dello stream (fine della partizione dell'argomento). Man mano che i consumatori recuperano il ritardo, MSK Replicator ridurrà automaticamente il ritardo.

La latenza di replica è elevata o continua ad aumentare

Di seguito sono riportate alcune cause comuni dell'elevata latenza di replica.

1. Verifica di disporre del numero corretto di partizioni nei cluster MSK di origine e di destinazione. Un numero di partizioni troppo basso o elevato può influire sulle prestazioni. Per indicazioni sulla scelta del numero di partizioni, consulta la sezione Best practice per l'utilizzo del replicatore MSK. La tabella seguente mostra il numero minimo di partizioni consigliato per ottenere la velocità di trasmissione effettiva desiderata con il replicatore MSK.

   Velocità di trasmissione effettiva e numero minimo consigliato di partizioni

Velocità di trasmissione effettiva (MB/s)	Numero minimo di partizioni necessarie
50	167
100	334
250	833
500	1666
1000	3333

2. Verifica di disporre di una capacità di lettura e scrittura sufficiente nei cluster MSK di origine e di destinazione per supportare il traffico di replica. Il replicatore MSK funge da consumatore per il cluster di origine (uscita) e da produttore per il cluster di destinazione (ingresso). Pertanto, è necessario fornire la capacità del cluster per supportare il traffico di replica oltre al resto del traffico sui cluster. Consulta la sezione ??? per indicazioni sul dimensionamento dei cluster MSK.
3. La latenza di replica può variare per i cluster MSK in diverse coppie di AWS regioni di origine e destinazione, a seconda della distanza geografica dei cluster l'uno dall'altro. Ad esempio, la latenza di replica è in genere inferiore quando si esegue la replica tra cluster nelle regioni Europa (Irlanda) ed Europa (Londra) rispetto alla replica tra cluster nelle regioni Europa (Irlanda) e Asia Pacifico (Sydney).
4. Assicurati che il replicatore non subisca limitazioni a causa delle quote eccessivamente aggressive impostate sui cluster di origine o di destinazione. Puoi utilizzare la ThrottleTime metrica fornita da MSK Replicator in Amazon CloudWatch per vedere il tempo medio, in millisecondi, in cui una richiesta è stata limitata dai broker del tuo cluster. source/target Se questo parametro è superiore a 0, è necessario modificare le quote Kafka per ridurre la limitazione della larghezza di banda della

rete in modo che il replicatore possa recuperare il ritardo. Per informazioni sulla gestione delle quote Kafka per il replicatore, consulta la pagina Gestione della velocità di trasmissione effettiva del replicatore MSK utilizzando le quote Kafka.

5. ReplicationLatency e potrebbe aumentare quando una regione si deteriora. MessageLag AWS Utilizza Dashboard AWS Service Health per verificare la presenza di un evento del servizio MSK nella regione in cui si trova il cluster MSK primario. Se si verifica un evento di servizio, è possibile reindirizzare temporaneamente le operazioni di lettura e scrittura dell'applicazione all'altra regione.

Risoluzione dei problemi relativi agli errori di MSK Replicator utilizzando la metrica ReplicatorFailure

La ReplicatorFailure metrica consente di monitorare e rilevare i problemi di replica in MSK Replicator. Un valore diverso da zero di questa metrica indica in genere un problema di errore di replica, che potrebbe derivare dai seguenti fattori:

- limitazioni alla dimensione dei messaggi
- violazioni dell'intervallo di data e ora
- registra i problemi relativi alla dimensione del batch

Se la ReplicatorFailure metrica riporta un valore diverso da zero, segui questi passaggi per risolvere il problema.

> **Note**
>
> Per ulteriori informazioni su questa metrica, consulta Parametri del replicatore MSK.

1. Configura un client in grado di connettersi al cluster MSK di destinazione e che disponga degli strumenti CLI di Apache Kafka. Per informazioni sulla configurazione del client e dello strumento Kafka CLI, vedere. Connect a un cluster Amazon MSK Provisioned
2. Aprire la console Amazon MSK a https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.

   Successivamente, esegui queste operazioni:

   a. Ottenete il file di MSK Replicator e il cluster MSK di destinazione. ARNs

b. Ottieni gli endpoint del broker del cluster MSK di destinazione. Utilizzerai questi endpoint nei passaggi seguenti.

3. Esegui i seguenti comandi per esportare l'ARN di MSK Replicator e gli endpoint del broker ottenuti nel passaggio precedente.

   Assicuratevi di sostituire i valori segnaposto per **<** *ReplicatorARN* **>**, **< >** e **<** *BootstrapServerString ConsumerConfigFile* **>** utilizzati negli esempi seguenti con i valori effettivi.

```
export TARGET_CLUSTER_SERVER_STRING=<BootstrapServerString>
```

```
export REPLICATOR_ARN=<ReplicatorARN>
```

```
export CONSUMER_CONFIG_FILE=<ConsumerConfigFile>
```

4. Nella tua <*path-to-your-kafka-installation*>/bin directory, procedi come segue:

   a. Salva il seguente script e assegnagli un nome**query-replicator-failure-message.sh**.

```
#!/bin/bash

# Script: Query MSK Replicator Failure Message
# Description: This script queries exceptions from AWS MSK Replicator status
 topics
# It takes a replicator ARN and bootstrap server as input and searches for
 replicator exceptions
# in the replicator's status topic, formatting and displaying them in a
 readable manner
#
# Required Arguments:
#   --replicator-arn: The ARN of the AWS MSK Replicator
#   --bootstrap-server: The Kafka bootstrap server to connect to
#   --consumer.config: Consumer config properties file
# Usage Example:
#   ./query-replicator-failure-message.sh ./query-replicator-failure-message.sh
 --replicator-arn <replicator-arn> --bootstrap-server <bootstrap-server> --
consumer.config <consumer.config>

print_usage() {
```

```
  echo "USAGE: $0 ./query-replicator-failure-message.sh --replicator-arn
 <replicator-arn> --bootstrap-server <bootstrap-server> --consumer.config
 <consumer.config>"
  echo "--replicator-arn <String: MSK Replicator ARN>      REQUIRED: The ARN of
 AWS MSK Replicator."
  echo "--bootstrap-server <String: server to connect to>  REQUIRED: The Kafka
 server to connect to."
  echo "--consumer.config <String: config file>            REQUIRED: Consumer
 config properties file."
  exit 1
}

# Initialize variables
replicator_arn=""
bootstrap_server=""
consumer_config=""

# Parse arguments
while [[ $# -gt 0 ]]; do
  case "$1" in
    --replicator-arn)
      if [ -z "$2" ]; then
        echo "Error: --replicator-arn requires an argument."
        print_usage
      fi
      replicator_arn="$2"; shift 2 ;;
    --bootstrap-server)
      if [ -z "$2" ]; then
        echo "Error: --bootstrap-server requires an argument."
        print_usage
      fi
      bootstrap_server="$2"; shift 2 ;;
    --consumer.config)
      if [ -z "$2" ]; then
        echo "Error: --consumer.config requires an argument."
        print_usage
      fi
      consumer_config="$2"; shift 2 ;;
    *) echo "Unknown option: $1"; print_usage ;;
  esac
done

# Check for required arguments
```

```
if [ -z "$replicator_arn" ] || [ -z "$bootstrap_server" ] || [ -z
 "$consumer_config" ]; then
  echo "Error: --replicator-arn, --bootstrap-server, and --consumer.config are
 required."
  print_usage
fi

# Extract replicator name and suffix from ARN
replicator_arn_suffix=$(echo "$replicator_arn" | awk -F'/' '{print $NF}')
replicator_name=$(echo "$replicator_arn" | awk -F'/' '{print $(NF-1)}')
echo "Replicator name: $replicator_name"

# List topics and find the status topic
topics=$(./kafka-topics.sh --command-config client.properties --list --
bootstrap-server "$bootstrap_server")
status_topic_name="__amazon_msk_replicator_status_${replicator_name}_
${replicator_arn_suffix}"

# Check if the status topic exists
if echo "$topics" | grep -Fq "$status_topic_name"; then
  echo "Found replicator status topic: '$status_topic_name'"
  ./kafka-console-consumer.sh --bootstrap-server "$bootstrap_server" --
consumer.config "$consumer_config" --topic "$status_topic_name" --from-
beginning | stdbuf -oL grep "Exception" | stdbuf -oL sed -n 's/.*Exception:\(.*
\) Topic: \([^,]*\), Partition: \([^\]*\).*/ReplicatorException:\1 Topic: \2,
 Partition: \3/p'
else
  echo "No topic matching the pattern '$status_topic_name' found."
fi
```

b. Esegui questo script per interrogare i messaggi di errore di MSK Replicator.

```
<path-to-your-kafka-installation>/bin/query-replicator-failure-message.sh --
replicator-arn $REPLICATOR_ARN --bootstrap-server $TARGET_CLUSTER_SERVER_STRING
 --consumer.config $CONSUMER_CONFIG_FILE
```

Questo script restituisce tutti gli errori con i relativi messaggi di eccezione e le partizioni tematiche interessate. È possibile utilizzare queste informazioni sulle eccezioni per mitigare gli errori come descritto in. Errori comuni di MSK Replicator e relative soluzioni Poiché l'argomento contiene tutti i messaggi di errore cronologici, avviate l'indagine utilizzando l'ultimo messaggio. Di seguito è riportato un esempio di messaggio di errore.

```
ReplicatorException: The request included a message larger than the max message
 size the server will accept. Topic: test, Partition: 1
```

Errori comuni di MSK Replicator e relative soluzioni

L'elenco seguente descrive alcuni degli errori di MSK Replicator che potrebbero verificarsi e come mitigarli.

Dimensione del messaggio superiore a max.request.size

Causa

Questo errore si verifica quando MSK Replicator non riesce a replicare i dati perché la dimensione del singolo messaggio supera i 10 MB. Per impostazione predefinita, MSK Replicator replica messaggi di dimensioni fino a 10 MB.

Di seguito è riportato un esempio di questo tipo di messaggio di errore.

```
ReplicatorException: The message is 20635370 bytes when serialized which is larger
 than 10485760, which is the value of the max.request.size configuration. Topic:
 test, Partition: 1
```

Soluzione

Riduci le dimensioni dei singoli messaggi nel tuo argomento. Se non riesci a farlo, segui queste istruzioni per richiedere un aumento del limite.

Dimensione del messaggio superiore alla dimensione massima dei messaggi accettata dal server

Causa

Questo errore si verifica quando la dimensione del messaggio supera la dimensione massima del messaggio del cluster di destinazione.

Di seguito è riportato un esempio di questo tipo di messaggio di errore.

```
ReplicatorException: The request included a message larger than the max message size
 the server will accept. Topic: test, Partition: 1
```

Soluzione

Aumenta la `max.message.bytes` configurazione sul cluster di destinazione o sull'argomento del cluster di destinazione corrispondente. Imposta la `max.message.bytes` configurazione del cluster di destinazione in modo che corrisponda alla dimensione massima dei messaggi non compressi. Per informazioni su questa operazione, consulta max.message.bytes.

Il timestamp non è compreso nell'intervallo

Causa

Questo errore si verifica perché il timestamp del singolo messaggio non rientra nell'intervallo consentito del cluster di destinazione.

Di seguito è riportato un esempio di questo tipo di messaggio di errore.

```
ReplicatorException: Timestamp 1730137653724 of message with offset 0 is out of
 range. The timestamp should be within [1730137892239, 1731347492239] Topic: test,
 Partition: 1
```

Soluzione

Aggiorna la `message.timestamp.before.max.ms` configurazione del cluster di destinazione per consentire la ricezione di messaggi con timestamp precedenti. Per informazioni su questa operazione, consulta message.timestamp.before.max.ms.

Batch di registrazione troppo grande

Causa

Questo errore si verifica perché la dimensione del batch di record supera la dimensione del segmento impostata per l'argomento nel cluster di destinazione. MSK Replicator supporta una dimensione batch massima di 1 MB.

Di seguito è riportato un esempio di questo tipo di messaggio di errore.

```
ReplicatorException: The request included message batch larger than the configured
 segment size on the server. Topic: test, Partition: 1
```

Soluzione

Affinché Replicator possa procedere senza errori, la configurazione segment.bytes del cluster di destinazione deve avere almeno le dimensioni del batch (1 MB). Aggiornate il file segment.bytes

del cluster di destinazione in modo che sia almeno 1048576 (1 MB). Per informazioni su questa operazione, consulta segment.bytes.

> **Note**
>
> Se la ReplicatorFailure metrica continua a emettere valori diversi da zero dopo aver applicato queste soluzioni, ripeti la procedura di risoluzione dei problemi finché la metrica non emette un valore pari a zero.

Best practice per l'utilizzo del replicatore MSK

Questa sezione illustra le best practice e le strategie di implementazione più comuni per l'utilizzo di Amazon MSK Replicator.

Argomenti

- Gestione della velocità di trasmissione effettiva del replicatore MSK utilizzando le quote Kafka
- Impostazione del periodo di conservazione dei cluster

Gestione della velocità di trasmissione effettiva del replicatore MSK utilizzando le quote Kafka

Poiché il replicatore MSK funge da consumatore per il cluster di origine, la replica può causare la limitazione della larghezza di banda della rete di altri consumatori sul cluster di origine. L'entità della limitazione della larghezza di banda della rete dipende dalla capacità di lettura disponibile sul cluster di origine e dalla velocità di trasmissione effettiva dei dati da replicare. Ti consigliamo di fornire una capacità identica per i cluster di origine e di destinazione e di tenere conto della velocità di trasmissione effettiva di replica nel calcolo della capacità necessaria.

È inoltre possibile impostare quote Kafka per il replicatore sui cluster di origine e di destinazione per controllare la capacità che il replicatore MSK può utilizzare. Si consiglia di specificare una quota di larghezza di banda della rete. Una quota di larghezza di banda della rete definisce una soglia di velocità di byte, espressa in byte al secondo, per uno o più client che condividono una quota. Questa quota è definita per singolo broker.

Segui questi passaggi per applicare una quota.

1. Recupera la stringa del server di bootstrap per il cluster di origine. Consultare Ottieni i broker bootstrap per un cluster Amazon MSK.
2. Recupera il ruolo di esecuzione del servizio (SER) utilizzato dal replicatore MSK. Questo è il SER che hai utilizzato per una richiesta `CreateReplicator`. Puoi anche estrarre il SER dalla DescribeReplicator risposta di un Replicator esistente.
3. Utilizzando gli strumenti della CLI di Kafka, esegui il seguente comando sul cluster di origine.

```
./kafka-configs.sh --bootstrap-server <source-cluster-bootstrap-server> --alter --
add-config 'consumer_byte_
rate=<quota_in_bytes_per_second>' --entity-type users --entity-name
 arn:aws:sts::<customer-account-id>:assumed-role/<ser-role-name>/<customer-account-
id> --command-config <client-properties-for-iam-auth></programlisting>
```

4. Dopo aver eseguito il comando precedente, verifica che il parametro `ReplicatorThroughput` non superi la quota impostata.

Nota che se riutilizzi un ruolo di esecuzione del servizio tra più replicatori MSK, questi sono tutti soggetti a questa quota. Se desideri mantenere quote separate per il replicatore, utilizza ruoli di esecuzione del servizio separati.

Per ulteriori informazioni sull'utilizzo dell'autenticazione IAM di MSK con le quote, consulta la pagina Multi-tenancy Apache Kafka clusters in Amazon MSK with IAM access control and Kafka Quotas – Part 1.

> **Warning**
>
> L'impostazione di un valore consumer_byte_rate estremamente basso può causare comportamenti inaspettati da parte del replicatore MSK.

Impostazione del periodo di conservazione dei cluster

È possibile impostare il periodo di conservazione dei log sia per i cluster MSK assegnati sia per quelli serverless. Il periodo di conservazione consigliato è di 7 giorni. Vedi Modifiche alla configurazione del cluster o Configurazione del cluster MSK Serverless supportata.

Integrazioni di MSK

Questa sezione fornisce riferimenti alle AWS funzionalità che si integrano con Amazon MSK.

Argomenti

Connettore Amazon Athena per Amazon MSK

Il connettore Amazon Athena per Amazon MSK consente ad Amazon Athena di eseguire query SQL sugli argomenti di Apache Kafka. Utilizza questo connettore per visualizzare gli argomenti di Apache Kafka come tabelle e i messaggi come righe in Athena.

Per ulteriori informazioni, consulta la pagina Amazon Athena MSK Connector nella Guida per l'utente di Amazon Athena.

Inserimento di dati in streaming di Amazon Redshift per Amazon MSK

Amazon Redshift supporta l'importazione dei dati in streaming da Amazon MSK. La funzionalità di importazione dei dati in streaming di Amazon Redshift fornisce l'importazione a bassa latenza e ad alta velocità dei dati in streaming da Amazon MSK in una vista materializzata di Amazon Redshift. Poiché non richiede la gestione temporanea dei dati in Amazon S3, Amazon Redshift può importare dati in streaming a una latenza inferiore e a un costo di archiviazione ridotto. Puoi configurare l'importazione dei dati in streaming di Amazon Redshift su un cluster Amazon Redshift utilizzando le istruzioni SQL per autenticarti e connetterti a un argomento Amazon MSK.

Per ulteriori informazioni, consulta la pagina Streaming ingestion nella Guida per gli sviluppatori di database di Amazon Redshift.

Integrazione Firehose per Amazon MSK

Amazon MSK si integra con Firehose per fornire una soluzione serverless e senza codice per distribuire flussi dai cluster Apache Kafka ai data lake Amazon S3. Firehose è un servizio di streaming di estrazione, trasformazione e caricamento (ETL) che legge i dati dagli argomenti di Amazon MSK Kafka, esegue trasformazioni come la conversione in Parquet e aggrega e scrive i dati su Amazon S3. Con pochi clic dalla console, puoi configurare uno stream Firehose da leggere da un argomento di Kafka e inviarlo a una posizione S3. Non richiede la scrittura di codice, applicazioni di connessione né provisioning di risorse. Firehose si ridimensiona automaticamente in base alla quantità di dati pubblicati sull'argomento Kafka e paghi solo per i byte acquisiti da Kafka.

Per ulteriori informazioni su questa funzionalità, consulta le seguenti risorse.

- Scrittura su Kinesis Data Firehose utilizzando Amazon MSK - Amazon Kinesis Data Firehose nella Amazon Data Firehose Developer Guide
- Post del blog: Amazon MSK Introduces Managed Data Delivery from Apache Kafka to Your Data Lake
- Laboratorio: consegna ad Amazon S3 tramite Firehose

Accedi ad Amazon EventBridge Pipes tramite la console Amazon MSK

Amazon EventBridge Pipes collega le sorgenti alle destinazioni. Le pipe sono destinate point-to-point alle integrazioni tra sorgenti e destinazioni supportate, con supporto per trasformazioni e arricchimenti avanzati. EventBridge Le pipe offrono un modo altamente scalabile per connettere il tuo cluster Amazon MSK a AWS servizi come Step Functions, Amazon SQS e API Gateway, nonché ad applicazioni SaaS (Software as a Service) di terze parti come Salesforce.

Per configurare una pipe, scegli l'origine, aggiungi filtri opzionali, definisci l'arricchimento opzionale e scegli la destinazione per i dati dell'evento.

Nella pagina dei dettagli di un cluster Amazon MSK, puoi visualizzare le pipe che utilizzano quel cluster come origine. Da lì, puoi anche:

- Avvia la console per visualizzare i dettagli delle pipe. EventBridge
- Avvia la EventBridge console per creare una nuova pipe con il cluster come origine.

Per ulteriori informazioni sulla configurazione di un cluster Amazon MSK come sorgente pipe, consulta Amazon Managed Streaming for Apache Kafka cluster come sorgente nella Amazon User Guide. EventBridge Per ulteriori informazioni su Pipes in generale, consulta EventBridge Pipes. EventBridge

Per accedere alle EventBridge pipe per un determinato cluster Amazon MSK

1. Apri la console Amazon MSK e seleziona Cluster.
2. Seleziona un cluster.
3. Nella pagina Dettagli del cluster, scegli la scheda Integrazione.

   La scheda Integrazione include un elenco di tutte le pipe attualmente configurate per utilizzare il cluster selezionato come origine, tra cui:

   - nome della pipe
   - stato corrente
   - destinazione della pipe
   - quando la pipe è stata modificata l'ultima volta
4. Gestisci le pipe per il tuo cluster Amazon MSK come desideri:

   Accesso a dettagli aggiuntivi su una pipe

   - Scegli la pipe.

     Verrà avviata la pagina dei dettagli di Pipe della EventBridge console.

   Creazione di una nuova pipe

   - Scegli Connetti il cluster Amazon MSK alla pipe.

     Verrà avviata la pagina Create pipe della EventBridge console, con il cluster Amazon MSK specificato come origine pipe. Per ulteriori informazioni, consulta Creating an EventBridge pipe nella Amazon EventBridge User Guide.

   - Puoi creare una pipe per un cluster anche dalla pagina Cluster. Seleziona il cluster e, dal menu Azioni, seleziona Create EventBridge Pipe.

Utilizzo di Kafka Streams con i broker MSK Express e MSK Serverless

Kafka Streams supporta trasformazioni stateless e stateful. Le trasformazioni stateful, come count, aggregate o join, utilizzano operatori che memorizzano il loro stato in argomenti interni di Kafka. Inoltre, alcune trasformazioni stateless come groupBy o repartition memorizzano i risultati in argomenti interni di Kafka. Per impostazione predefinita, Kafka Streams nomina questi argomenti interni in base all'operatore corrispondente. Se questi argomenti non esistono, Kafka Streams crea argomenti Kafka interni. Per creare gli argomenti interni, Kafka Streams codifica la configurazione segment.bytes e la imposta su 50 MB. MSK Provisioned with Express brokers e MSK Serverless protegge alcune configurazioni degli argomenti, tra cui segment.size, durante la creazione degli argomenti. Pertanto, un'applicazione Kafka Streams con trasformazioni stateful non riesce a creare gli argomenti interni utilizzando i broker MSK Express o MSK Serverless.

Per eseguire tali applicazioni Kafka Streams sui broker MSK Express o MSK Serverless, è necessario creare personalmente gli argomenti interni. A tale scopo, è necessario innanzitutto identificare e assegnare un nome agli operatori di Kafka Streams, che richiedono argomenti. Quindi, create i corrispondenti argomenti interni di Kafka.

> **Note**
>
> - È buona norma denominare manualmente gli operatori in Kafka Streams, specialmente quelli che dipendono da argomenti interni. Per informazioni sulla denominazione degli operatori, vedere Naming Operators in a Kafka Streams DSL Application nella documentazione di Kafka Streams.
> - Il nome dell'argomento interno per una trasformazione stateful dipende dall'applicazione Kafka Streams e dal nome `application.id` dell'operatore stateful,. `application.id-statefuloperator_name`

Argomenti

- Creazione di un'applicazione Kafka Streams utilizzando i broker MSK Express o MSK Serverless

Creazione di un'applicazione Kafka Streams utilizzando i broker MSK Express o MSK Serverless

Se l'applicazione Kafka Streams è `application.id` impostata sum`sk-streams-processing`, è possibile creare un'applicazione Kafka Streams utilizzando i broker MSK Express o MSK Serverless. Per fare ciò, utilizzate l'`count()`operatore, che richiede un argomento interno con il nome. Ad esempio `msk-streams-processing-count-store`.

Per creare un'applicazione Kafka Streams, procedi come segue:

Argomenti

- Identifica e assegna un nome agli operatori
- Create gli argomenti interni
- (Facoltativo) Controlla il nome dell'argomento
- Esempi di operatori di denominazione

Identifica e assegna un nome agli operatori

1. Identifica i processori con stato utilizzando le trasformazioni Stateful nella documentazione di Kafka Streams.

   Alcuni esempi di processori con stato includono, o. `count` `aggregate` `join`
2. Identifica i processori che creano argomenti per il ripartizionamento.

   L'esempio seguente contiene un'`count()`operazione che richiede uno stato.

   ```
   var stream =
       paragraphStream
           .groupByKey()
           .count()
           .toStream();
   ```
3. Per assegnare un nome all'argomento, aggiungete un nome per ogni processore stateful. In base al tipo di processore, la denominazione viene effettuata da una classe di denominazione diversa. Ad esempio, `count()` l'operazione è un'operazione di aggregazione. Pertanto, ha bisogno della `Materialized` classe.

Per informazioni sulle classi di denominazione per le operazioni stateful, vedi Conclusioni nella documentazione di Kafka Streams.

L'esempio seguente imposta il nome dell'operatore per l'utilizzo della classe. `count()` `count-store` `Materialized`

```
var stream =
    paragraphStream
        .groupByKey()
        .count(Materialized.<String, Long, KeyValueStore<Bytes, byte[]>>as("count-
store")  // descriptive name for the store
            .withKeySerde(Serdes.String())
            .withValueSerde(Serdes.Long()))
        .toStream();
```

Create gli argomenti interni

I prefissi di Kafka Streams `application.id` ai nomi degli argomenti interni, dove sono definiti dall'utente. `application.id` Ad esempio `application.id-internal_topic_name`. Gli argomenti interni sono normali argomenti di Kafka e puoi crearli utilizzando le informazioni disponibili in o dell'API Kafka. Crea un argomento di Apache Kafka `AdminClient`

A seconda del caso d'uso, è possibile utilizzare le politiche di pulizia e conservazione predefinite di Kafka Streams o personalizzarne i valori. Li definisci in e. `cleanup.policy` `retention.ms`

L'esempio seguente crea gli argomenti con l'`AdminClient`API e li imposta `application.id` su**`msk-streams-processing`**.

```
try (AdminClient client = AdminClient.create(configs.kafkaProps())) {
    Collection<NewTopic> topics = new HashSet<>();
    topics.add(new NewTopic("msk-streams-processing-count-store", 3, (short) 3));
    client.createTopics(topics);
}
```

Dopo aver creato gli argomenti nel cluster, l'applicazione Kafka Streams può utilizzare l'`msk-streams-processing-count-store`argomento per l'operazione. `count()`

(Facoltativo) Controlla il nome dell'argomento

Puoi usare il descrittore topografico per descrivere la topologia del tuo stream e visualizzare i nomi degli argomenti interni. L'esempio seguente mostra come eseguire il descrittore di topologia.

```
final StreamsBuilder builder = new StreamsBuilder();
Topology topology = builder.build();
System.out.println(topology.describe());
```

L'output seguente mostra la topologia del flusso per l'esempio precedente.

```
Topology Description:
Topologies:
   Sub-topology: 0
    Source: KSTREAM-SOURCE-0000000000 (topics: [input_topic])
      --> KSTREAM-AGGREGATE-0000000001
    Processor: KSTREAM-AGGREGATE-0000000001 (stores: [count-store])
      --> KTABLE-TOSTREAM-0000000002
      <-- KSTREAM-SOURCE-0000000000
    Processor: KTABLE-TOSTREAM-0000000002 (stores: [])
      --> KSTREAM-SINK-0000000003
      <-- KSTREAM-AGGREGATE-0000000001
    Sink: KSTREAM-SINK-0000000003 (topic: output_topic)
      <-- KTABLE-TOSTREAM-0000000002
```

Per informazioni su come utilizzare il descrittore di topologia, vedere Naming Operators in a Kafka Streams DSL Application nella documentazione di Kafka Streams.

Esempi di operatori di denominazione

Questa sezione fornisce alcuni esempi di operatori di denominazione.

Esempio di operatore di denominazione per groupByKey ()

```
groupByKey() -> groupByKey(Grouped.as("kafka-stream-groupby"))
```

Esempio di operatore di denominazione per il normale count ()

```
normal count() -> .count(Materialized.<String, Long, KeyValueStore<Bytes,
 byte[]>>as("kafka-streams-window")  // descriptive name for the store
    .withKeySerde(Serdes.String())
```

```
    .withValueSerde(Serdes.Long()))
```

Esempio di operatore di denominazione per windowed count ()

```
windowed count() -> .count(Materialized.<String, Long, WindowStore<Bytes,
 byte[]>>as("kafka-streams-window")  // descriptive name for the store
    .withKeySerde(Serdes.String())
    .withValueSerde(Serdes.Long()))
```

Esempio di operatore di denominazione per windowowed suppressed ()

```
windowed suppressed() ->
Suppressed<Windowed> suppressed = Suppressed
        .untilWindowCloses(Suppressed.BufferConfig.unbounded())
        .withName("kafka-suppressed");
    .suppress(suppressed)
```

Progetti di incorporamento vettoriale in tempo reale

Amazon MSK (Managed Streaming for Apache Kafka) supporta i blueprint di Amazon Managed Service for Apache Flink per generare incorporamenti vettoriali utilizzando Amazon Bedrock, semplificando il processo per creare applicazioni AI in tempo reale basate su dati contestuali. up-to-date Il modello MSF semplifica il processo di incorporazione dei dati più recenti delle pipeline di streaming Amazon MSK nei modelli di intelligenza artificiale generativa, eliminando la necessità di scrivere codice personalizzato per integrare flussi di dati in tempo reale, database vettoriali e modelli linguistici di grandi dimensioni.

Puoi configurare il blueprint MSF per generare continuamente incorporamenti vettoriali utilizzando i modelli di incorporamento di Bedrock, quindi indicizzare tali incorporamenti in OpenSearch Service per i relativi flussi di dati Amazon MSK. Ciò consente di combinare il contesto dei dati in tempo reale con i potenti modelli di linguaggio di grandi dimensioni di Bedrock per generare risposte IA accurate senza scrivere codice personalizzato. up-to-date Puoi anche scegliere di migliorare l'efficienza del recupero dei dati utilizzando il supporto integrato per le tecniche di suddivisione in blocchi dei dati di una libreria open source LangChain, che supporta input di alta qualità per l'ingestione di modelli. Il blueprint gestisce l'integrazione e l'elaborazione dei dati tra MSK, il modello di incorporamento scelto, e il OpenSearch vector store, consentendovi di concentrarvi sulla creazione di applicazioni di intelligenza artificiale, anziché sulla gestione dell'integrazione sottostante.

I progetti di incorporamento vettoriale in tempo reale sono disponibili nelle seguenti Regioni: AWS

- Virginia settentrionale - us-east-1
- Ohio - us-east-2
- Oregon - us-west-2
- Mumbai - ap-south-1
- Seul - ap-northeast-2
- Singapore - ap-southeast-1
- Sydney - ap-southeast-2
- Tokyo - ap-northeast-1
- Canada centrale - ca-central-1
- Francoforte - eu-central-1
- Irlanda - eu-west-1
- Londra - eu-west-2
- Parigi - eu-west-3
- San Paolo - sa-east-1

Argomenti

Registrazione e osservabilità

Tutti i log e le metriche per i blueprint di incorporamento vettoriale in tempo reale possono essere abilitati utilizzando i log. CloudWatch

Tutte le metriche disponibili per una normale applicazione MSF e Amazon Bedrock possono monitorare l'applicazione e i parametri di Bedrock.

Esistono due metriche aggiuntive per il monitoraggio delle prestazioni di generazione degli incorporamenti. Queste metriche fanno parte del nome dell' EmbeddingGeneration operazione in. CloudWatch

- BedrockTitanEmbeddingTokenCount: monitora il numero di token presenti in una singola richiesta a Bedrock.

- BedrockEmbeddingGenerationLatencyMs: riporta il tempo impiegato per inviare e ricevere una risposta da Bedrock per la generazione di incorporamenti in millisecondi.

Per OpenSearch il servizio, puoi utilizzare i seguenti parametri:

- OpenSearch Metriche di raccolta serverless: consulta Monitoring OpenSearch Serverless with Amazon CloudWatch nella Amazon OpenSearch Service Developer Guide.
- OpenSearch metriche assegnate: consulta Monitoring OpenSearch cluster metrics with Amazon CloudWatch nella Amazon OpenSearch Service Developer Guide.

Note prima di abilitare i blueprint di incorporamento vettoriale in tempo reale

l'applicazione di servizio gestito per Apache Flink supporterà unicamente testo non strutturato o dati JSON nel flusso di input;

Sono supportate due modalità di elaborazione degli input:

- Quando i dati di input sono testo non strutturato, viene incorporato l'intero messaggio di testo. Il DB vettoriale contiene il testo originale e l'incorporamento generato.
- Quando i dati di input sono in formato JSON, l'applicazione offre la possibilità di configurare e specificare una o più chiavi all'interno del valore dell'oggetto JSON da utilizzare per il processo di incorporamento. Se è presente più di una chiave, tutte le chiavi vengono vettorializzate insieme e indicizzate nel DB vettoriale. Il DB vettoriale conterrà il messaggio originale e l'incorporamento generato.

Generazione di incorporamento: l'applicazione supporta tutti i modelli di incorporamento del testo forniti esclusivamente da Bedrock.

Persistenza in Vector DB Store: l'applicazione utilizza un OpenSearch cluster esistente (fornito o Serverless) nell'account del cliente come destinazione per la persistenza dei dati incorporati. Quando usi Opensearch Serverless per creare un indice vettoriale, usa sempre il nome del campo vettoriale. `embedded_data`

Analogamente ai blueprint MSF, è necessario gestire l'infrastruttura per eseguire il codice associato al blueprint di incorporamento vettoriale in tempo reale.

Analogamente a MSF Blueprints, una volta creata, un'applicazione MSF deve essere avviata esclusivamente nell' AWS account utilizzando la console o la CLI. AWS non avvierà l'applicazione

MSF per te. È necessario chiamare l' StartApplication API (tramite CLI o console) per far funzionare l'applicazione.

Spostamento dei dati tra account: l'applicazione non consente di spostare dati tra flussi di input e destinazioni vettoriali che risiedono in account diversi. AWS

Implementa un modello di vettorizzazione dei dati in streaming

Questo argomento descrive come implementare un modello di vettorizzazione dei dati in streaming.

Implementa un modello di vettorizzazione dei dati in streaming

1. Assicurati che le seguenti risorse siano configurate correttamente:
   - Cluster MSK con provisioning o serverless con uno o più argomenti contenenti dati.
2. Bedrock Setup: accesso al modello Bedrock desiderato. I modelli Bedrock attualmente supportati sono:
   - Titan Embeddings G1 - Text
   - Amazon Titan Text Embeddings V2
   - Amazon Titan Multimodal Embeddings G1
   - Cohere Embed English
   - Cohere Embed Multilingual
3. AWS OpenSearch collezione:
   - È possibile utilizzare una raccolta di OpenSearch servizi predisposta o Serverless.
   - la raccolta di OpenSearch servizi deve avere almeno un indice;
   - Se prevedi di utilizzare una raccolta OpenSearch Serverless, assicurati di creare una raccolta di ricerca vettoriale. Per informazioni dettagliate su come configurare un indice vettoriale, consultate Prerequisiti per il proprio archivio vettoriale per una knowledge base. Per ulteriori informazioni sulla vettorizzazione, consulta la spiegazione delle funzionalità del database vettoriale di Amazon OpenSearch Service.

     > **Note**
     >
     > Quando crei un indice vettoriale, devi usare il nome del campo vettoriale. `embedded_data`

- Se prevedi di utilizzare una raccolta OpenSearch Provisioned, devi aggiungere il ruolo dell'applicazione MSF (che contiene la politica di accesso Opensearch) creato dal blueprint, come utente principale della tua raccolta. OpenSearch Inoltre, verificate che la politica di accesso in OpenSearch sia impostata su «Consenti» azioni. Ciò è necessario per consentire un controllo preciso degli accessi.
- Facoltativamente, puoi abilitare l'accesso alla OpenSearch dashboard per visualizzare i risultati. Consultate per abilitare il controllo degli accessi a grana fine.

4. Accedi utilizzando un ruolo che consente aws: CreateStack permessi.
5. Vai alla dashboard della console MSF e seleziona Crea applicazione di streaming.
6. In Scegli un metodo per configurare l'applicazione di elaborazione dello stream seleziona Usa un progetto.
7. Seleziona Real-time AI application blueprint dal menu a discesa Blueprint.
8. Fornisci le configurazioni desiderate. Consultare Creare configurazioni di pagina.
9. Seleziona Deploy Blueprint per avviare una distribuzione. CloudFormation
10. Una volta completata la CloudFormation distribuzione, vai all'applicazione Flink distribuita. Controlla le proprietà di runtime dell'applicazione.
11. È possibile scegliere di modificare/aggiungere proprietà di runtime all'applicazione. Vedi Runtime Properties Configuration per i dettagli sulla configurazione di queste proprietà.

> **Note**
>
> Nota:
> Se utilizzi OpenSearch provisioned, assicurati di aver abilitato il controllo degli accessi a grana fine.
> Se il cluster fornito è privato, aggiungilo `https://` all'URL dell'endpoint OpenSearch VPC Provisioned e modificalo in modo che punti `sink.os.endpoint` a questo endpoint.
> Se il cluster fornito è pubblico, assicurati che l'applicazione MSF possa accedere a Internet. Per ulteriori informazioni, consulta >>>>>> express-brokers-publication-merge type="documentation» url="managed- flink/latest/java/vpc -internet.html ">Accesso a Internet e ai servizi per un'applicazione Managed Service for Apache Flink connessa a VPC.

12. Quando sei soddisfatto di tutte Run le configurazioni, seleziona. L'applicazione inizierà a funzionare.

13. Invia messaggi nel tuo cluster MSK.
14. Vai al cluster Opensearch e vai alla OpenSearch dashboard.
15. Nella dashboard, seleziona Discover nel menu a sinistra. Dovresti vedere i documenti persistenti insieme ai relativi incorporamenti vettoriali.
16. Consultate Lavorare con le raccolte di ricerca vettoriale per scoprire come utilizzare i vettori memorizzati nell'indice.

Creare configurazioni di pagina

Questo argomento descrive le configurazioni di creazione di pagine a cui fare riferimento quando si specificano le configurazioni per i blueprint di applicazioni AI in tempo reale.

Nome applicazione

Campo esistente in MSF, dai un nome alla tua applicazione.

Cluster MSK

Seleziona il cluster MSK creato durante l'installazione dall'elenco a discesa.

Argomenti

Aggiungi il nome dell'argomento o degli argomenti che hai creato durante la configurazione.

Tipo di dati del flusso di input

Scegliete String se volete fornire un input di stringa allo stream MSK.

Scegliete JSON se l'input nel flusso MSK è JSON. Nelle chiavi JSON incorporate, scrivi i nomi dei campi nel tuo JSON di input il cui valore desideri inviare a Bedrock per generare gli incorporamenti.

Modello Bedrock Embedrock

Selezionane uno dall'elenco. Assicurati di avere accesso al modello che hai scelto, altrimenti lo stack potrebbe fallire. Vedi Aggiungere o rimuovere l'accesso ai modelli Amazon Bedrock Foundation.

OpenSearch cluster

Seleziona il cluster creato dall'elenco a discesa.

OpenSearch nome dell'indice vettoriale

Seleziona l'indice vettoriale che hai creato nel passaggio precedente.

Quota di Amazon MSK

Hai Account AWS delle quote predefinite per Amazon MSK. Salvo diversa indicazione, ogni quota per account è specifica per regione all'interno del tuo paese. Account AWS

Argomenti

Richiesta di un aumento della quota in Amazon MSK

Puoi richiedere un aumento della quota per ogni regione utilizzando la console Service Quotas o un caso di supporto. AWS CLI Se una quota regolabile non è disponibile nella console Service Quotas, utilizza il caso AWS Support Center Console per creare un caso di aumento della quota di servizio.

Il supporto potrebbe approvare, rifiutare o approvare parzialmente le tue richieste di aumento delle quote. Gli aumenti non vengono concessi immediatamente e possono essere necessari alcuni giorni prima che abbiano effetto.

Per richiedere un aumento utilizzando la console Service Quotas

1. Apri la console Service Quotas all'indirizzo https://console.aws.amazon.com/servicequotas/.
2. Dalla barra di navigazione, nella parte superiore dello schermo, seleziona una regione.
3. Nel riquadro di navigazione a sinistra, scegliere Servizi AWS.
4. Nella casella Trova servizi**`msk`**, digita e scegli Amazon Managed Streaming for Apache Kafka (MSK).
5. In Quote di servizio, scegli il nome della quota per la quale desideri richiedere un aumento. Ad esempio, **`Number of brokers per account`**.
6. Scegli Richiedi aumento a livello di account.
7. Per Aumentare il valore della quota, inserisci un nuovo valore di quota.

8. Scegli Richiedi.
9. (Facoltativo) Per visualizzare le richieste in sospeso o risolte di recente nella console, scegli Dashboard nel riquadro di navigazione a sinistra. Per le richieste in sospeso, scegliere lo stato della richiesta per aprire la ricevuta della richiesta. Lo stato iniziale di una richiesta è Pending (In attesa). Dopo la modifica dello stato in Quota richiesta, vedrai il numero del caso in Support. Scegli il numero del caso per aprire il ticket della tua richiesta.

Per ulteriori informazioni, incluso come utilizzare AWS CLI o richiedere un aumento della quota, SDKs vedere Richiedere un aumento della quota nella Guida per l'utente di Service Quotas.

Quota di broker Amazon MSK Standard

La tabella seguente descrive le quote per i broker Standard.

Dimensione	Quota	Note
Broker per account	90	Per richiedere una quota più elevata, vai alla console Service Quotas.
Broker per cluster	30 per i cluster ZooKeeper basati, 60 per KRaft i cluster basati	Per richiedere una quota più elevata, vai alla console Service Quotas.
Spazio di archiviazione minimo per broker	1 GiB	
Spazio di archiviazione massimo per broker	16384 GiB	
Numero massimo di connessioni TCP per broker (controllo degli accessi IAM)	3000	Per aumentare questo limite, puoi modificare la proprietà `listener.name.client_iam.max.connections` o la proprietà di `listener.name.client_iam_public.max.`

Dimensione	Quota	Note
		`connections` configurazione utilizzando l'`AlterConfig` API Kafka o lo strumento `kafka-configs.sh`. È importante notare che impostare una delle due proprietà su un valore elevato può comportare l'indisponibilità.
Velocità massima di connessioni TCP per broker (IAM)	100 al secondo (dimensioni delle istanze M5 e M7g) 4 al secondo (dimensione dell'istanza t3)	Per gestire i tentativi di connessione non riusciti, puoi impostare il parametro di configurazione `reconnect.backoff.ms` sul lato client. Ad esempio, se desideri che un client riprovi a connettersi dopo 1 secondo, imposta su. `reconnect.backoff.ms` `1000` Per ulteriori informazioni, consulta la sezione reconnect.backoff.ms nella documentazione di Apache Kafka.
Numero massimo di connessioni TCP per broker (non IAM)	N/D	MSK non impone limiti di connessione per l'autenticazione non IAM. È necessario monitorare altre metriche come l'utilizzo della CPU e della memoria per assicurarsi di non sovraccaricare il cluster a causa di connessioni eccessive.

Dimensione	Quota	Note
Configurazioni per account	100	Per richiedere una quota più elevata, vai alla console Service Quotas. Per aggiornare la configura zione o la versione Apache Kafka di un cluster MSK, assicurati innanzitutto che il numero di partizioni per broker sia inferiore ai limiti descritti in Dimensionamento corretto del cluster: numero di partizioni per broker standard.
Revisioni della configurazione per account	50	

Quota del broker Amazon MSK Express

La tabella seguente descrive le quote per i broker Express.

Dimensione	Quota	Note
Broker per account	90	Per richiedere una quota più elevata, vai alla console Service Quotas.
Broker per cluster	30	Per richiedere una quota più elevata, vai alla console Service Quotas.
Spazio di archiviazione massimo	Illimitato	

Dimensione	Quota	Note
Numero massimo di connessioni TCP per broker (controllo degli accessi IAM)	3000	Per aumentare il limite di connessione, modifica una delle seguenti proprietà di configurazione utilizzando l' AlterConfig API Kafka o lo strumento kafka-configs.sh: • `listener.name.client_iam.max.connections` • `listener.name.client_iam_public.max.connections` L'impostazione di queste proprietà su un valore elevato potrebbe comportare l'indisponibilità del cluster.
Velocità massima di connessioni TCP per broker (IAM)	100 al secondo	Per gestire i tentativi di connessione non riusciti, puoi impostare il parametro di configurazione `reconnect.backoff.ms` sul lato client. Ad esempio, se desideri che un client riprovi a connettersi dopo 1 secondo, imposta su `reconnect.backoff.ms` . `1000` Per ulteriori informazioni, consulta la sezione reconnect.backoff.ms nella documentazione di Apache Kafka.

Dimensione	Quota	Note
Numero massimo di connessioni TCP per broker (non IAM)	N/D	MSK non impone limiti di connessione per l'autenticazione non IAM. È tuttavia necessario monitorare altre metriche come l'utilizzo della CPU e della memoria per assicurarsi di non sovraccaricare il cluster a causa di connessioni eccessive.
Configurazioni per account	100	Per richiedere una quota più elevata, vai alla console Service Quotas. Per aggiornare la configurazione o la versione Apache Kafka di un cluster MSK, assicurati innanzitutto che il numero di partizioni per broker sia inferiore ai limiti descritti in Dimensioni corrette del cluster: numero di partizioni per broker Express.
Revisioni della configurazione per account	50	
Ingresso massimo per broker	Consigliato: 15,6 - 500,0 MBps	In base alla dimensione dell'istanza.
Uscita massima per broker	Consigliato: 31,2 - 1000.0 MBps	In base alla dimensione dell'istanza.

Argomenti

- Limiti di velocità di trasmissione del broker Express in base alle dimensioni del broker
- Quota di partizione Express Broker

Limiti di velocità di trasmissione del broker Express in base alle dimensioni del broker

La tabella seguente elenca il limite massimo e consigliato per quanto riguarda l'accelerazione del throughput in ingresso e in uscita per broker di diverse dimensioni. In questa tabella, la velocità effettiva consigliata è rappresentata come Prestazioni sostenute, che è la soglia fino alla quale le applicazioni non subiranno alcun peggioramento delle prestazioni. Se operi oltre questi limiti in entrambe le dimensioni, potresti ottenere una maggiore velocità effettiva, ma potresti anche riscontrare un peggioramento delle prestazioni. La quota massima è la soglia oltre la quale il cluster limiterà il traffico. read/write Le tue applicazioni non saranno in grado di funzionare oltre questa soglia.

Dimensioni istanza	Prestazioni sostenute (MBps) per l'ingresso	Quota massima (MBps) per l'ingresso	Prestazioni sostenute (MBps) per l'uscita	Quota massima (MBps) per l'uscita
express.m 7g.large	15,6	23,4	31,2	58,5
express.m 7g.xlarge	31.2	46,8	62,5	117
express.m7g. 2xlarge	62,5	93,7	125	234,2
express.m7g. 4xlarge	124,9	187,5	249,8	468,7
express.m 7g.8xlarge	250	375	500	937,5
express.m7 g. 12 x grande	375	562,5	750	1406,2
express.m7g. 16 x grande	500	750	1000	1875

Quota di partizione Express Broker

La tabella seguente mostra il numero consigliato di partizioni (incluse le repliche leader e follower) per ogni broker Express. Non è possibile superare il numero massimo di partizioni indicato nella tabella seguente per ogni broker Express.

Per informazioni sulle migliori pratiche da prendere in considerazione durante l'assegnazione delle partizioni ai broker Express, consulta. Dimensioni corrette del cluster: numero di partizioni per broker Express

Dimensioni del broker	Numero consigliato di partizioni (incluse le repliche leader e follower) per broker	Numero massimo di partizioni per broker
`express.m7g.large`	1000	1500
`express.m7g.xlarge`	1000	2000
`express.m7g.2xlarge`	2500	4000
`express.m7g.4xlarge`	6000	8000
`express.m7g.8xlarge`	12000	16000
`express.m7g.12xlarge`	16000	24000
`express.m7g.16xlarge`	20000	32000

Quote del replicatore MSK

- Un massimo di 15 replicatori MSK per account.
- MSK Replicator replica solo fino a 750 argomenti in ordine ordinato. Se è necessario replicare più argomenti, si consiglia di creare un Replicator separato. Vai alla console Service Quotas, se hai bisogno di supporto per più di 750 argomenti per Replicator. È possibile monitorare il numero di argomenti replicati utilizzando la metrica "»TopicCount.
- Una velocità di trasmissione effettiva di ingresso massima di 1 GB al secondo per replicatore MSK. Richiedi una quota più alta tramite la console Service Quotas.

- Dimensione del record MSK Replicator: una dimensione massima del record di 10 MB (message.max.bytes). Richiedi una quota più alta tramite la console Service Quotas.

Quota di MSK Serverless

Le quote specificate nella tabella seguente sono per cluster, salvo diversa indicazione.

> **Note**
>
> In caso di problemi con i limiti delle quote di servizio, crea una richiesta di supporto con il tuo caso d'uso e il limite richiesto.

Dimensione	Quota	Risultato della violazione della quota
Velocità di trasmissione effettiva massima in ingresso	200 MBps	Rallentamento con limitazione della larghezza di banda della rete prolungata in risposta
Velocità di trasmissione effettiva massima in uscita	400 MBps	Rallentamento con limitazione della larghezza di banda della rete prolungata in risposta
Durata massima di conservazione	Illimitato	N/D
Numero massimo di connessioni client	3000	Chiusura della connessione
Numero massimo di tentativi di connessione	100 al secondo	Chiusura della connessione
Dimensione massima del messaggio	8 MiB	La richiesta fallisce con ErrorCode: INVALID_REQUEST

Dimensione	Quota	Risultato della violazione della quota
Velocità massima di richieste	15.000 al secondo	Rallentamento con limitazione della larghezza di banda della rete prolungata in risposta
Frequenza massima delle richieste di gestione degli argomenti APIs	2 al secondo	Rallentamento con limitazione della larghezza di banda della rete prolungata in risposta
Numero massimo di byte di recupero per richiesta	55 MB	La richiesta ha esito negativo con ErrorCode: INVALID_REQUEST
Numero massimo di gruppi di consumatori	500	JoinGroup la richiesta fallisce
Numero massimo di partizioni (leader)	2.400 per argomenti non compattati. 120 per argomenti compattati. Per richiedere un adeguamento della quota di servizio, crea un caso di supporto con il tuo caso d'uso e il limite richiesto.	La richiesta ha esito negativo con ErrorCode: INVALID_REQUEST
Velocità massima di creazione ed eliminazione delle partizioni	250 in 5 minuti	La richiesta ha esito negativo con: THROUGHPUT_QUOTA_EXCEEDED ErrorCode
Velocità di trasmissione effettiva massima in ingresso per partizione	5 MBps	Rallentamento con limitazione della larghezza di banda della rete prolungata in risposta
Velocità di trasmissione effettiva massima in uscita per partizione	10 MBps	Rallentamento con limitazione della larghezza di banda della rete prolungata in risposta

Dimensione	Quota	Risultato della violazione della quota
Dimensione massima della partizione (per argomenti compatti)	250 GB	La richiesta non riesce con ErrorCode: THROUGHPUT_QUOTA_EXCEEDED
Numero massimo di client per cluster serverless VPCs	5	
Numero massimo di cluster serverless per account	10. Per richiedere un adeguamento della quota di servizio, crea un caso di supporto con il tuo caso d'uso e il limite richiesto.	

Quota di MSK Connect

- Fino a 100 plug-in personalizzati.
- Fino a 100 configurazioni di worker.
- Fino a 60 worker di connessione. Se un connettore è configurato in modo da avere una capacità con dimensionamento automatico, MSK Connect utilizza il numero massimo di worker che il connettore è configurato per avere al fine di calcolare la quota per l'account.
- Fino a 10 worker per connettore.

Per richiedere una quota più elevata per MSK Connect, accedete alla console Service Quotas.

Cronologia del documento Guida per gli sviluppatori di Amazon MSK

Nella tabella seguente sono descritte le modifiche importanti apportate alla Guida per gli sviluppatori di Amazon MSK.

Ultimo aggiornamento della documentazione: 25 giugno 2024

Modifica	Descrizione	Data
StorageUsed metrica per i broker Express	Amazon MSK ora include una nuova metrica di livello DEFAULT, `StorageUsed` per i broker Express, che offre visibilità a livello di cluster sul consumo totale di storage, escluse le repliche. Per ulteriori informazioni, consulta Monitoraggio del livello DEFAULT per i broker Express.	24-07-2025
Numero elevato di partizioni per i broker Amazon MSK Express	Amazon MSK ha lanciato un numero elevato di partizioni per i broker Express. Per ulteriori informazioni, consulta Express broker Partition Quota.	21-07-2025
Nuovi parametri di Amazon MSK Connect	MSK Connect ha aggiunto due nuove metriche, oltre a misurare `SinkConsumerByteRate` `SourceProducerByteRate` le velocità di trasmissione dei dati dei connettori. Per ulteriori	2025-06-30

Modifica	Descrizione	Data
	informazioni, vedere Monitoring MSK Connect.	
MSK Provisioned Guida introduttiva alla revisione degli argomenti	Gli argomenti introduttivi di MSK Provisioned sono stati completamente ristrutturati per migliorare l'esperienza utente, il flusso dei contenuti e la leggibilità. Gli argomenti rivisti includono anche una documentazione dettagliata di tutte le opzioni della console, comprese le modalità di archiviazione, i metodi di autenticazione e i livelli di monitoraggio, con chiare linee guida decisionali. Per ulteriori informazioni, consulta Inizia a usare Amazon MSK.	2025-06-28
Supporto per il broker Express su Apache Kafka versione 3.8.x	Amazon MSK ora supporta i broker Express su Apache Kafka versione 3.8.x. Per ulteriori informazioni, consulta i broker Amazon MSK Express.	2025-06-05

Modifica	Descrizione	Data
Nuove informazioni sulla risoluzione dei problemi di Amazon MSK Replicator	La Amazon Managed Streaming for Apache Kafka Developer Guide ora include una documentazione completa per la risoluzione dei problemi di MSK Replicator con script di diagnostica e procedure dettagliate di mitigazione degli errori. Per ulteriori informazioni, consulta Risolvere gli errori di MSK Replicator utilizzando la metrica. ReplicatorFailure	2025-05-09
Rinnovi dei certificati senza interruzioni	Amazon MSK ha lanciato rinnovi dei certificati senza interruzioni per i broker Express per eliminare i tempi di inattività della manutenzione durante gli aggiornamenti obbligatori dei certificati per 13 mesi. Per ulteriori informazioni, consulta la sezione Crittografia Amazon MSK.	2025-05-05
Support per Apache Kafka versione 4.0.x	Amazon MSK ora supporta la versione 4.0.x di Apache Kafka. Per ulteriori informazioni, consulta Versioni supportate di Apache Kafka.	2025-05-02

Modifica	Descrizione	Data
Lancio di Amazon MSK Connect nelle regioni della Cina	MSK Connect è ora disponibile in tutte le regioni della Cina: Cina (Pechino) e Cina (Ningxia).	2025-04-10
Support per Apache Kafka versione 3.9.x	Amazon MSK ora supporta la versione 3.9.x di Apache Kafka. Per informazioni, consulta Versioni supportate di Apache Kafka.	21 aprile 2025
Connettore sink Amazon EventBridge Kafka per MSK Connect	La Amazon Managed Streaming for Apache Kafka Developer Guide ora include un argomento completo che descrive come EventBridge usare il connettore Kafka sink con MSK Connect. Per ulteriori informazioni, vedere Configurazione del connettore EventBridge Kafka sink per MSK Connect.	2025-03-28
Aggiornamento delle quote dei broker Express	La Amazon Managed Streaming for Apache Kafka Developer Guide ora include informazioni sui limiti di velocità effettiva in ingresso e in uscita per i broker Express. Per ulteriori informazioni, consulta la pagina relativa alla quota del broker Amazon MSK Express.	2025-03-06

Modifica	Descrizione	Data
Support per Apache Kafka versione 3.8.x	Amazon MSK ora supporta la versione 3.8.x di Apache Kafka. Per ulteriori informazioni, consulta Versioni supportate di Apache Kafka.	2025-02-20
Revisione della data di fine del supporto per la versione 3.4.0 di Amazon MSK	La nuova data di fine del supporto per la versione 3.4.0 di Apache Kafka è il 4 agosto 2025. Per ulteriori informazioni, consulta Versioni supportate di Apache Kafka.	18/02/2025
UpdateConnector Avvio dell'API per modificare le configurazioni dei connettori MSK Connect esistenti	Amazon MSK ora include l'UpdateConnectorAPI per modificare le configurazioni dei connettori MSK Connect esistenti eliminando la necessità di creare nuovi connettori. Inoltre, è stata aggiunta la documentazione DescribeConnectorOperatione il monitoraggio delle operazioni ListConnectorOperations APIs di aggiornamento dei connettori e il mantenimento degli audit trail cronologici delle modifiche alla configurazione.	12/01/2025

Modifica	Descrizione	Data
AWS PrivateLink documentazione degli endpoint VPC dell'interfaccia	La Guida per gli sviluppatori di Amazon Managed Streaming for Apache Kafka AWS PrivateLink include ora la documentazione sugli endpoint VPC dell'interfaccia. Per ulteriori informazioni, consulta Usare Amazon MSK APIs con endpoint VPC di interfaccia.	2024-12-18
Lancio della versione MSK Connect 3.7.x	MSK Connect ora supporta la versione 3.7.x. Per ulteriori informazioni, vedere Understand MSK Connect.	2024-12-18
Aggiunta la funzionalità di broker Express. Argomenti della Guida per gli sviluppatori riorganizzati.	MSK supporta i broker Standard e New Express.	2024-11-6
Aggiunta la funzionalità Graviton upgrade in place.	È possibile aggiornare le dimensioni del cluster broker da M5 o T3 a M7g o da M7g a M5.	2024-6-25
3.4.0 Annunciata la data di fine del supporto.	La data di fine del supporto per la versione 3.4.0 di Apache Kafka è il 17 giugno 2025.	2024-6-24

Modifica	Descrizione	Data
Aggiunta la funzionalità di rimozione del broker.	È possibile ridurre la capacità di storage e di elaborazione del cluster assegnato rimuovendo set di broker, senza alcun impatto sulla disponibilità, rischi di durabilità dei dati o interruzione delle applicazioni di streaming dei dati.	2024-5-16
`WriteDataIdempotently` aggiunto a AWSMSKReplicator ExecutionRole	WriteDataIdempotently è stata aggiunta l'autorizzazione alla AWSMSKReplicator Execution Role policy per supportare la replica dei dati tra cluster MSK.	2024-5-16
Broker Graviton M7g rilasciati in Brasile e Bahrein.	Amazon MSK ora supporta la disponibilità nelle regioni del Sud America (sa-east-1, San Paolo) e del Medio Oriente (me-south-1, Bahrein) dei broker M7g che utilizzano processori Graviton (processori personalizzati basati su ARM creati da Amazon Web Services). AWS	2024-2-07
Rilascia i broker Graviton M7g nella regione della Cina	Amazon MSK ora supporta la disponibilità nella regione della Cina dei broker M7g che utilizzano processori AWS Graviton (processori personalizzati basati su ARM creati da Amazon Web Services).	2024-01-11

Modifica	Descrizione	Data
Politica di supporto della versione di Amazon MSK Kafka	È stata aggiunta una spiegazione della politica di supporto della versione di Kafka supportata da Amazon MSK. Per ulteriori informazioni, consulta le versioni di Apache Kafka.	2023-12-08
Nuova politica dei ruoli di esecuzione del servizio per supportare Amazon MSK Replicator.	Amazon MSK ha aggiunto una nuova `AWSMSKReplicatorExecutionRole` policy per supportare Amazon MSK Replicator. Per ulteriori informazioni, consulta l'argomento relativo alle policy gestite da AWS : `AWSMSKReplicatorExecutionRole` .	2023-12-06
Supporto M7g Graviton	Amazon MSK ora supporta i broker M7g che utilizzano processori AWS Graviton (processori personalizzati basati su ARM creati da Amazon Web Services).	2023-11-27

Modifica	Descrizione	Data
Replicatore Amazon MSK	Il replicatore Amazon MSK è una nuova funzionalità che puoi utilizzare per replicare i dati tra cluster Amazon MSK. Amazon MSK Replicator include un aggiornamento della policy di Amazon MSKFull Access. Per ulteriori informazioni, consulta l'argomento relativo alle policy gestite da AWS : `AmazonMSKFullAccess`.	28/09/2023
Sono state aggiornate le best practice IAM.	Guida aggiornata per l'allineamento alle best practice IAM. Per ulteriori informazioni, consulta Best practice per la sicurezza in IAM.	-08
Aggiornamenti del ruolo collegato ai servizi per supportare la connettività privata multi-VPC	Amazon MSK ora include aggiornamenti dei ruoli AWSService RoleForKafka collegati ai servizi per gestire le interfacce di rete e gli endpoint VPC nel tuo account che rendono i broker di cluster accessibili ai clienti nel tuo VPC. Amazon MSK utilizza le autorizzazioni per `DescribeVpcEndpoints`, `ModifyVpcEndpoint` e `DeleteVpcEndpoints`. Per ulteriori informazioni, consulta Ruoli collegati ai servizi per Amazon MSK.	1-08

Modifica	Descrizione	Data
Supporto per Apache Kafka 2.7.2	Amazon MSK ora supporta Apache Kafka versione 2.7.2. Per ulteriori informazioni, consulta Versioni di Apache Kafka supportate.	2021-12-21
Supporto per Apache Kafka 2.6.3	Amazon MSK ora supporta Apache Kafka versione 2.6.3. Per ulteriori informazioni, consulta Versioni di Apache Kafka supportate.	2021-12-21
Versione preliminare di MSK Serverless	MSK Serverless è una nuova funzionalità che è possibile utilizzare per creare cluster serverless. Per ulteriori informazioni, consulta MSK Serverless.	2021-11-29
Supporto per Apache Kafka 2.8.1	Amazon MSK ora supporta Apache Kafka versione 2.8.1. Per ulteriori informazioni, consulta Versioni di Apache Kafka supportate.	2021-09-30
MSK Connect	MSK Connect è una nuova funzionalità che è possibile utilizzare per creare e gestire i connettori Apache Kafka. Per ulteriori informazioni, consulta Comprendere MSK Connect.	2021-09-16

Modifica	Descrizione	Data
Supporto per Apache Kafka 2.7.1	Amazon MSK ora supporta Apache Kafka versione 2.7.1. Per ulteriori informazioni, consulta Versioni di Apache Kafka supportate.	2021-05-25
Supporto per Apache Kafka 2.8.0	Amazon MSK ora supporta Apache Kafka versione 2.8.0. Per ulteriori informazioni, consulta Versioni di Apache Kafka supportate.	2021-04-28
Supporto per Apache Kafka 2.6.2	Amazon MSK ora supporta Apache Kafka versione 2.6.2. Per ulteriori informazioni, consulta Versioni di Apache Kafka supportate.	2021-04-28
Supporto per l'aggiornamento del tipo di broker	È ora possibile modificare il tipo di broker per un cluster esistente. Per ulteriori informazioni, consulta Aggiornamento delle dimensioni del broker del cluster Amazon MSK.	2021-01-21
Supporto per Apache Kafka 2.6.1	Amazon MSK ora supporta Apache Kafka versione 2.6.1. Per ulteriori informazioni, consulta Versioni di Apache Kafka supportate.	2021-01-19

Modifica	Descrizione	Data
Supporto per Apache Kafka 2.7.0	Amazon MSK ora supporta Apache Kafka versione 2.7.0. Per ulteriori informazioni, consulta Versioni di Apache Kafka supportate.	2020-12-29
Indisponibilità di nuovi cluster con Apache Kafka versione 1.1.1	Non è più possibile creare un nuovo cluster Amazon MSK con la versione 1.1.1 di Apache Kafka. Tuttavia, se disponi di cluster MSK esistenti che eseguono Apache Kafka versione 1.1.1, puoi continuare a utilizzare tutte le funzionalità attualmen te supportate su tali cluster esistenti. Per ulteriori informazi oni, consulta Versioni di Apache Kafka.	2020-11-24
Parametri relativi al ritardo dei consumatori	Ora Amazon MSK offre parametri che permettono di monitorare il ritardo dei consumatori. Per ulteriori informazioni, consulta Monitora un cluster Amazon MSK Provisioned.	2020-11-23
Supporto per Cruise Control	Amazon MSK ora supporta il LinkedIn Cruise Control. Per ulteriori informazioni, consulta Usa LinkedIn il Cruise Control per Apache Kafka con Amazon MSK.	2020-11-17

Modifica	Descrizione	Data
Supporto per Apache Kafka 2.6.0	Amazon MSK ora supporta Apache Kafka versione 2.6.0. Per ulteriori informazioni, consulta Versioni di Apache Kafka supportate.	2020-10-21
Supporto per Apache Kafka 2.5.1	Amazon MSK ora supporta Apache Kafka versione 2.5.1. Con la versione 2.5.1 di Apache Kafka, Amazon MSK supporta la crittografia in transito tra client ed endpoint. ZooKeeper Per ulteriori informazioni, consulta Versioni di Apache Kafka supportate.	30/09/2020
Espansione automatica dell'applicazione	Puoi configurare Streaming gestito da Amazon per Apache Kafka per espandere automaticamente l'archiviazione del tuo cluster in risposta a un incremento dell'utilizzo. Per ulteriori informazioni, consulta Scalabilità automatica per i cluster.	2020-09-17
Supporto per la sicurezza di nome utente e password	Amazon MSK ora supporta l'accesso ai cluster utilizzando nome utente e password. Amazon MSK archivia le credenziali in AWS Secrets Manager. Per ulteriori informazioni, consulta Autenticazione SASL/SCRAM.	2020-09-17

Modifica	Descrizione	Data
Supporto per l'aggiornamento della versione Apache Kafka di un cluster Amazon MSK	Puoi aggiornare la versione di Apache Kafka di un cluster MSK esistente.	2020-05-28
Supporto per nodi broker T3.Small	Amazon MSK ora supporta la creazione di cluster con broker di tipo Amazon EC2 T3.small.	2020-04-08
Supporto per Apache Kafka 2.4.1.	Amazon MSK ora supporta Apache Kafka versione 2.4.1.	2020-04-02
Supporto per i log del broker di streaming	Amazon MSK ora può trasmettere i log dei broker a CloudWatch Logs, Amazon S3 e Amazon Data Firehose. Firehose può, a sua volta, consegnare questi log alle destinazioni supportate, come Service. OpenSearch	2020-02-25
Supporto per Apache Kafka 2.3.1.	Amazon MSK ora supporta Apache Kafka versione 2.3.1.	19-12-2019
Monitoraggio aperto	Amazon MSK ora supporta il monitoraggio aperto con Prometheus.	04-12-2019
Supporto per Apache Kafka 2.2.1.	Amazon MSK ora supporta Apache Kafka versione 2.2.1.	31-07-2019
Disponibilità generale	Le nuove caratteristiche includono il supporto di tagging, l'autenticazione, la crittografia TLS, le configura zioni e la possibilità di aggiornare lo storage broker.	30-05-2019

Modifica	Descrizione	Data
Supporto per Apache Kafka 2.1.0.	Amazon MSK ora supporta Apache Kafka versione 2.1.0.	05-02-2019

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.