aws

Entwicklerhandbuch

Amazon Managed Streaming für Apache Kafka

Amazon Managed Streaming für Apache Kafka: Entwicklerhandbuch

Table of Contents

Willkommen beim Entwicklerhandbuch für Amazon MSK

Willkommen beim Entwicklerhandbuch für Amazon MSK. Die folgenden Themen erleichtern Ihnen den Einstieg in dieses Handbuch anhand dessen, was Sie erreichen möchten.

- Erstellen Sie einen Amazon-MSK-Cluster, indem Sie dem Erste Schritte mit Amazon MSK-Tutorial folgen.
- Tauchen Sie tiefer in die Funktionalität von Amazon MSK in Amazon MSK: Funktionsweise ein.
- Führen Sie Apache Kafka aus, ohne die Cluster-Kapazität verwalten und skalieren zu müssen, mit MSK Serverless.
- Verwenden Sie MSK Connect, um Daten zu und von Ihrem Apache-Kafka-Cluster zu streamen.
- Wird verwendetMSK-Replikator, um Daten zuverlässig über Amazon MSK-Cluster in verschiedenen oder derselben AWS Region (en) zu replizieren.

Highlights, weitere Produktdetails und Preise finden Sie auf der Serviceseite für Amazon MSK.

Was ist Amazon MSK?

Amazon Managed Streaming for Apache Kafka (Amazon MSK) ist ein vollständig verwalteter Service, mit dem Sie Anwendungen erstellen und ausführen können, die Apache Kafka zum Verarbeiten von Streaming-Daten verwenden. Amazon MSK stellt die Vorgänge auf Steuerebene bereit, z. B. zum Erstellen, Aktualisieren und Löschen von Clustern. Damit können Sie Apache Kafka-Operationen auf Datenebene verwenden, z. B. zum Erstellen und Nutzen von Daten. Es werden Open-Source-Versionen von Apache Kafka ausgeführt. Das bedeutet, dass vorhandene Anwendungen, Tools und Plugins von Partnern und der Apache Kafka-Community unterstützt werden, ohne dass Änderungen am Anwendungscode erforderlich sind. Sie können Amazon MSK verwenden, um Cluster zu erstellen, die sämtliche Apache-Kafka-Versionen verwenden, die unter the section called “Unterstützte Apache Kafka-Versionen” aufgeführt sind.

Diese Komponenten beschreiben die Architektur von Amazon MSK:

- Broker-Knoten – Wenn Sie einen Amazon-MSK-Cluster erstellen, geben Sie an, wie viele Broker-Knoten Amazon MSK in jeder Availability Zone erstellen soll. Das Minimum ist ein Broker pro Availability Zone. Jede Availability Zone hat ein eigenes VPC(Virtual Private Cloud)-Subnetz.

- ZooKeeper Knoten — Amazon MSK erstellt auch die ZooKeeper Apache-Knoten für Sie. Apache ZooKeeper ist ein Open-Source-Server, der eine äußerst zuverlässige verteilte Koordination ermöglicht.
- KraFT-Controller — Die Apache Kafka-Community hat KraFT entwickelt, um Apache ZooKeeper für die Metadatenverwaltung in Apache Kafka-Clustern zu ersetzen. Im KraFT-Modus werden Cluster-Metadaten innerhalb einer Gruppe von Kafka-Controllern, die Teil des Kafka-Clusters sind, und nicht knotenübergreifend verbreitet. ZooKeeper KraFT-Controller sind ohne zusätzliche Kosten für Sie enthalten und erfordern keine zusätzliche Einrichtung oder Verwaltung durch Sie.

  > **Note**
  >
  > Ab Apache Kafka Version 3.7.x auf MSK können Sie Cluster erstellen, die den KraFT-Modus anstelle des Modus verwenden. ZooKeeper

- Produzenten, Verbraucher und Themenersteller – Mit Amazon MSK können Sie Apache-Kafka-Vorgänge auf Datenebene verwenden, um Themen zu erstellen und Daten zu produzieren und zu verbrauchen.
- Cluster-Operationen Sie können die APIs AWS Management Console, die AWS Command Line Interface (AWS CLI) oder die APIs im SDK verwenden, um Operationen auf der Steuerungsebene auszuführen. Sie können beispielsweise einen Amazon-MSK-Cluster erstellen oder löschen, alle Cluster in einem Konto auflisten, die Eigenschaften eines Clusters anzeigen und die Anzahl und den Typ der Broker in einem Cluster aktualisieren.

Amazon MSK erkennt die häufigsten Ausfallszenarien und stellt sich automatisch wieder her, sodass Ihre Produzenten- und Verbraucher-Anwendungen ihre Schreib- und Lesevorgänge mit minimalen Auswirkungen fortsetzen können. Wenn Amazon MSK einen Broker-Fehler entdeckt, wird der fehlerhafte oder nicht erreichbaren Broker durch einen neuen Broker ersetzt. Darüber hinaus wird, soweit möglich, der Speicher des älteren Brokers wiederverwendet, um die von Apache Kafka zu replizierende Datenmenge zu verringern. Die Auswirkungen auf Ihre Verfügbarkeit sind auf den Zeitraum begrenzt, den Amazon MSK für die Erkennung und Wiederherstellung benötigt. Nach einer Wiederherstellung können Ihre Hersteller- und Verbraucheranwendungen weiterhin mit denselben Broker-IP-Adressen kommunizieren, die sie vor dem Ausfall verwendet haben.

Einrichten von Amazon MSK

Bevor Sie Amazon MSK zum ersten Mal verwenden, führen Sie die folgenden Aufgaben aus.

Aufgaben

- Melde dich an für AWS
- Bibliotheken und Tools herunterladen

Melde dich an für AWS

Wenn Sie sich für registrieren AWS, wird Ihr Amazon Web Services Services-Konto automatisch für alle Services angemeldet AWS, einschließlich Amazon MSK. Berechnet werden Ihnen aber nur die Services, die Sie nutzen.

Wenn Sie bereits ein AWS Konto haben, fahren Sie mit der nächsten Aufgabe fort. Wenn Sie kein AWS -Konto haben, führen Sie die folgenden Schritte zum Erstellen eines Kontos aus.

So registrieren Sie sich für ein Konto bei Amazon Web Services

1. Öffnen Sie https://portal.aws.amazon.com/billing/signup.
2. Folgen Sie den Online-Anweisungen.

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für einen anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird ein erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Aus Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich ist.

Bibliotheken und Tools herunterladen

Die folgenden Bibliotheken und Tools können Sie bei der Arbeit mit Amazon MSK unterstützen:

- Die AWS Command Line Interface (AWS CLI) unterstützt Amazon MSK. Das AWS CLI ermöglicht es Ihnen, mehrere Amazon Web Services von der Befehlszeile aus zu steuern und sie mithilfe von Skripten zu automatisieren. Führen Sie ein Upgrade AWS CLI auf die neueste Version durch,

um sicherzustellen, dass sie die in diesem Benutzerhandbuch dokumentierten Amazon MSK-Funktionen unterstützt. Ausführliche Anweisungen zum Aktualisieren der AWS CLI finden Sie unter Installieren der AWS Command Line Interface. Nachdem Sie das installiert haben AWS CLI, müssen Sie es konfigurieren. Informationen zur Konfiguration von finden Sie AWS CLI unter aws configure.

- Die API-Referenz zu Amazon Managed Streaming für Kafka dokumentiert die API-Vorgänge, die Amazon MSK unterstützt.
- Die Amazon Web Services SDKs für Go, Java, .NET JavaScript, Node.js, PHP, Python und Ruby enthalten Amazon MSK-Unterstützung und Beispiele.

Erste Schritte mit Amazon MSK

In diesem Tutorial sehen Sie ein Beispiel, wie Sie mithilfe von Metriken einen MSK-Cluster erstellen, Daten erzeugen und verbrauchen und den Zustand Ihres Clusters überwachen können. Dieses Beispiel zeigt nicht alle Optionen, die Sie auswählen können, wenn Sie einen MSK-Cluster erstellen. In verschiedenen Teilen dieses Tutorials wählen wir aus Gründen der Einfachheit die Standardoptionen. Dies bedeutet nicht, dass dies die einzigen Optionen sind, um einen MSK-Cluster oder Client-Instances einzurichten.

Themen

Schritt 1: Einen Amazon-MSK-Cluster erstellen

In diesem Schritt von Erste Schritte mit Amazon MSK erstellen Sie einen Amazon-MSK-Cluster.

Um einen Amazon MSK-Cluster mit dem zu erstellen AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon MSK-Konsole unter https://console.aws.amazon.com/msk/home?region=us-east-1#/home/.
2. Wählen Sie Cluster erstellen.
3. Lassen Sie für die Erstellungsmethode die Option Schnellerstellung ausgewählt. Mit der Option Schnellerstellung können Sie einen Cluster mit Standardeinstellungen erstellen.
4. Geben Sie unter Cluster-Name einen Namen für den Cluster ein. z. B. **`MSKTutorialCluster`**.
5. Wählen Sie für Allgemeine Cluster-Eigenschaften Bereitgestellt als Cluster-Typ.
6. Kopieren Sie aus der Tabelle unter Alle Cluster-Einstellungen die Werte der folgenden Einstellungen und speichern Sie sie, da Sie sie später in diesem Tutorial benötigen:

- VPC
- Subnetze
- Die mit der VPC verknüpften Sicherheitsgruppen

7. Wählen Sie Cluster erstellen.
8. Überprüfen Sie den Cluster-Status auf der Seite Cluster-Zusammenfassung. Der Status ändert sich von Erstellen auf Aktiv, wenn Amazon MSK den Cluster bereitstellt. Wenn der Status Active lautet, können Sie die Verbindung mit dem Cluster herstellen. Weitere Informationen zu Cluster-Status finden Sie unter Cluster-Status.

Nächster Schritt

Schritt 2: Erstellen einer IAM-Rolle

Schritt 2: Erstellen einer IAM-Rolle

In diesem Schritt führen Sie zwei Aufgaben aus. Die erste Aufgabe besteht darin, eine IAM-Richtlinie zu erstellen, die Zugriff auf die Erstellung von Themen auf dem Cluster und das Senden von Daten an diese Themen gewährt. Die zweite Aufgabe besteht darin, eine IAM-Rolle zu erstellen und ihr diese Richtlinie zuzuordnen. In einem späteren Schritt erstellen Sie einen Client-Computer, der diese Rolle übernimmt und sie verwendet, um ein Thema auf dem Cluster zu erstellen und Daten an dieses Thema zu senden.

So erstellen Sie eine IAM-Richtlinie, die es ermöglicht, Themen zu erstellen und in sie zu schreiben

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Richtlinien.
3. Wählen Sie Richtlinie erstellen aus.
4. Wählen Sie die Registerkarte JSON und ersetzen Sie dann den JSON-Code im Editor-Fenster durch den Folgenden.

   Ersetzen Sie *`Region`* durch den Code der AWS Region, in der Sie Ihren Cluster erstellt haben. Ersetzen Sie *`Konto-ID`* durch Ihre Konto-ID. Ersetzen Sie *`MSK TutorialCluster`* durch den Namen Ihres Clusters.

   ```
   {
   ```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:AlterCluster",
                "kafka-cluster:DescribeCluster"
            ],
            "Resource": [
                "arn:aws:kafka:region:Account-ID:cluster/MSKTutorialCluster/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:*Topic*",
                "kafka-cluster:WriteData",
                "kafka-cluster:ReadData"
            ],
            "Resource": [
                "arn:aws:kafka:region:Account-ID:topic/MSKTutorialCluster/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeGroup"
            ],
            "Resource": [
                "arn:aws:kafka:region:Account-ID:group/MSKTutorialCluster/*"
            ]
        }
    ]
}
```

Anleitungen zum Verfassen von sicheren Richtlinien finden Sie unter the section called “IAM-Zugriffssteuerung”.

5. Wählen Sie Next: Tags (Weiter: Tags) aus.
6. Klicken Sie auf Weiter: Prüfen.
7. Geben Sie für den Richtliniennamen einen aussagekräftigen Namen ein, z. B. msk-tutorial-policy.

8. Wählen Sie Richtlinie erstellen aus.

So erstellen Sie eine IAM-Rolle und fügen ihr die Richtlinie an

1. Wählen Sie im Navigationsbereich Rollen.
2. Wählen Sie Rolle erstellen.
3. Wählen Sie unter Häufige Anwendungsfälle die Option EC2 und dann Weiter: Berechtigungen.
4. Geben Sie in das Suchfeld den Namen der Richtlinie ein, die Sie zuvor für dieses Tutorial erstellt haben. Aktivieren Sie anschließend das Kontrollkästchen links neben der Richtlinie.
5. Wählen Sie Next: Tags (Weiter: Tags) aus.
6. Klicken Sie auf Weiter: Prüfen.
7. Geben Sie für den Rollennamen einen aussagekräftigen Namen ein, z. B. msk-tutorial-role.
8. Wählen Sie Rolle erstellen aus.

Nächster Schritt

Schritt 3: Einen Client-Computer erstellen

Schritt 3: Einen Client-Computer erstellen

In diesem Schritt von Erste Schritte mit Amazon MSK erstellen Sie einen Client-Computer. Sie verwenden diesen Client-Computer, um ein Thema zu erstellen, das Daten erzeugt und verwendet. Der Einfachheit halber erstellen Sie diesen Client-Computer in der VPC, die dem MSK-Cluster zugeordnet ist, sodass der Client problemlos eine Verbindung zum Cluster herstellen kann.

Erstellen eines Client-Computers

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie Launch Instances aus.
3. Geben Sie einen Namen für Ihren Client-Computer ein, z. B. **`MSKTutorialClient`**
4. Lassen Sie Amazon Linux 2 AMI (HVM) – Kernel 5.10, SSD Volume Type als Amazon Machine Image (AMI)-Typ ausgewählt.
5. Lassen Sie den t2.micro-Instance-Typ ausgewählt.

6. Wählen Sie unter Schlüsselpaar (Login) die Option Neues Schlüsselpaar erstellen. Geben Sie **`MSKKeyPair`** für den Schlüsselpaar-Namen ein und wählen Sie dann Schlüsselpaar herunterladen. Alternativ können Sie ein vorhandenes Schlüsselpaar verwenden.
7. Erweitern Sie den Abschnitt Erweiterte Details und wählen Sie die IAM-Rolle aus, die Sie in Schritt 2: Eine IAM-Rolle erstellen erstellt haben.
8. Wählen Sie Launch Instance (Instance starten) aus.
9. Klicken Sie auf View Instances (Instances anzeigen). Wählen Sie dann in der Spalte Sicherheitsgruppen die Sicherheitsgruppe, die Ihrer neuen Instance zugeordnet ist. Kopieren Sie die ID der Sicherheitsgruppe, und speichern Sie sie für später.
10. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
11. Wählen Sie im Navigationsbereich Security Groups (Sicherheitsgruppen) aus. Suchen Sie die Sicherheitsgruppe, deren ID Sie in the section called “Schritt 1: Erstellen eines Clusters” gespeichert haben.
12. Wählen Sie auf der Registerkarte Regeln für eingehenden Datenverkehr die Option Regeln für eingehenden Datenverkehr bearbeiten.
13. Wählen Sie Regel hinzufügen aus.
14. Wählen Sie in der neuen Regel All traffic (Gesamter Datenverkehr) in der Spalte Type (Typ). Wählen Sie im zweiten Feld in der Spalte Quelle die Sicherheitsgruppe des Client-Computers. Dies ist die Gruppe, deren Namen Sie gespeichert haben, nachdem Sie die Client-Computer-Instance gestartet haben.
15. Wählen Sie Save rules (Regeln speichern) aus. Jetzt kann die Sicherheitsgruppe des Clusters Datenverkehr akzeptieren, der von der Sicherheitsgruppe des Client-Computers stammt.

Nächster Schritt

Schritt 4: Ein Thema erstellen

Schritt 4: Ein Thema erstellen

In diesem Schritt von Erste Schritte mit Amazon MSK installieren Sie Apache-Kafka-Client-Bibliotheken und -Tools auf dem Client-Computer und erstellen dann ein Thema.

> **Warning**
>
> Die in diesem Tutorial verwendeten Versionsnummern von Apache Kafka sind nur Beispiele. Es wird empfohlen, dieselbe Version des Clients wie die MSK-Cluster-Version zu verwenden. In einer älteren Client-Version fehlen möglicherweise bestimmte Funktionen und kritische Bugfixes.

So finden Sie die Version Ihres MSK-Clusters

1. Rufen Sie https://eu-west-2.console.aws.amazon.com/msk/ auf
2. Wählen Sie den MSK-Cluster aus.
3. Notieren Sie sich die Version von Apache Kafka, die auf dem Cluster verwendet wird.
4. Ersetzen Sie die Amazon-MSK-Versionsnummern in diesem Tutorial durch die in Schritt 3 erhaltene Version.

Erstellen eines Themas auf dem Client-Computer

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich Instances aus. Aktivieren Sie dann das Kontrollkästchen neben dem Namen des Client-Computers, den Sie in Schritt 3: Einen Client-Computer erstellen erstellt haben.
3. Klicken Sie auf Actions (Aktionen) und anschließend auf Connect (Verbinden). Folgen Sie den Anweisungen in der Konsole, um eine Verbindung zum Client-Computer herzustellen.
4. Installieren Sie Java auf dem Client-Computer, indem Sie den folgenden Befehl ausführen:

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

5. Führen Sie den folgenden Befehl aus, um Apache Kafka herunterzuladen.

   ```
   wget https://archive.apache.org/dist/kafka/{YOUR MSK VERSION}/kafka_2.13-{YOUR MSK
    VERSION}.tgz
   ```

> **Note**
>
> Wenn Sie eine andere als die in diesem Befehl verwendete Spiegelsite verwenden möchten, können Sie eine andere auf der Apache-Website auswählen.

6. Führen Sie den folgenden Befehl in dem Verzeichnis aus, in das Sie im vorherigen Schritt die TAR-Datei heruntergeladen haben.

```
tar -xzf kafka_2.13-{YOUR MSK VERSION}.tgz
```

7. Wechseln Sie zum Verzeichnis `kafka_2.13-{YOUR MSK VERSION}/libs` und führen Sie dann den folgenden Befehl aus, um die Amazon-MSK-IAM-JAR-Datei herunterzuladen. Das Amazon-MSK-IAM-JAR ermöglicht dem Client-Computer den Zugriff auf den Cluster.

```
wget https://github.com/aws/aws-msk-iam-auth/releases/download/v1.1.1/aws-msk-iam-auth-1.1.1-all.jar
```

8. Wechseln Sie zum Verzeichnis `kafka_2.13-{YOUR MSK VERSION}/bin`. Kopieren Sie die folgenden Eigenschaften-Einstellungen und fügen Sie sie in eine neue Datei ein. Benennen Sie die Datei **`client.properties`** und speichern Sie sie.

```
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
```

9. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
10. Warten Sie, bis der Status Ihres Clusters Aktiv ist. Dies kann einige Minuten dauern. Wenn der Status Aktiv lautet, wählen Sie den Cluster-Namen aus. Dadurch gelangen Sie zu einer Seite mit der Cluster-Zusammenfassung.
11. Wählen Sie Client-Informationen anzeigen.
12. Kopieren Sie die Verbindungszeichenfolge für den privaten Endpunkt.

    Sie erhalten drei Endpunkte für jeden der Broker. Für den folgenden Schritt benötigen Sie nur einen Broker-Endpunkt.
13. Führen Sie den folgenden Befehl aus und ersetzen Sie *`BootstrapServerString`* durch einen der Broker-Endpunkte, die Sie im vorherigen Schritt abgerufen haben.

```
<path-to-your-kafka-installation>/bin/kafka-topics.sh --create --bootstrap-server
 BootstrapServerString --command-config client.properties --replication-factor 3 --
partitions 1 --topic MSKTutorialTopic
```

Wenn der Befehl erfolgreich ist, wird die folgende Meldung angezeigt: `Created topic MSKTutorialTopic.`

Nächster Schritt

Schritt 5: Produzieren und Verbrauchen von Daten

Schritt 5: Produzieren und Verbrauchen von Daten

In diesem Schritt von Erste Schritte mit Amazon MSK erstellen und verbrauchen Sie Daten.

Erstellen und Verbrauchen von Nachrichten

1. Führen Sie den folgenden Befehl aus, um einen Konsolenproduzenten zu starten. Ersetzen Sie *BootstrapServerString* durch die Klartext-Verbindungszeichenfolge, die Sie unter Thema erstellen abgerufen haben. Anweisungen zum Abrufen dieser Verbindungszeichenfolge finden Sie unter Bootstrap-Broker für einen Amazon-MSK-Cluster abrufen.

   ```
   <path-to-your-kafka-installation>/bin/kafka-console-producer.sh --
   broker-list BootstrapServerString --producer.config client.properties --
   topic MSKTutorialTopic
   ```

2. Geben Sie eine beliebige Nachricht ein, und drücken Sie Enter (Eingabetaste). Wiederholen Sie diesen Schritt zwei- oder dreimal. Jedes Mal, wenn Sie eine Zeile eingeben und Enter (Eingabetaste) drücken, wird diese Zeile als separate Nachricht an Ihren Apache Kafka-Cluster gesendet.
3. Lassen Sie die Verbindung zum Client-Computer geöffnet und öffnen Sie dann eine zweite separate Verbindung zu diesem Computer in einem neuen Fenster.
4. Ersetzen Sie im folgenden Befehl *BootstrapServerString* durch die Klartext-Verbindungszeichenfolge, die Sie zuvor gespeichert haben. Verwenden Sie dann Ihre zweite Verbindung zum Client-Computer, um mit dem folgenden Befehl einen Konsolen-Verbraucher zu erstellen.

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapServerString --consumer.config client.properties --
topic MSKTutorialTopic --from-beginning
```

Sie sehen die Nachrichten, die Sie zuvor eingegeben haben, als Sie den Konsolenproduzentenbefehl verwendet haben.

5. Geben Sie weitere Nachrichten in das Producer-Fenster ein und beobachten Sie, wie sie im Consumer-Fenster angezeigt werden.

Nächster Schritt

Schritt 6: Amazon CloudWatch zum Anzeigen von Amazon MSK-Metriken verwenden

Schritt 6: Amazon CloudWatch zum Anzeigen von Amazon MSK-Metriken verwenden

In diesem Schritt von Erste Schritte mit Amazon MSK sehen Sie sich die Amazon MSK-Metriken in Amazon an. CloudWatch

Um Amazon MSK-Metriken anzuzeigen in CloudWatch

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie die Registerkarte All Metrics (Alle Metriken) und dann AWS/Kafka.
4. Um Metriken auf Broker-Ebene anzuzeigen, wählen Sie Broker ID, Cluster Name (Broker-ID, Cluster-Name). Wählen Sie für Metriken auf Cluster-Ebene Cluster Name (Clustername).
5. (Optional) Wählen Sie im Diagrammbereich eine Statistik und einen Zeitraum aus, und erstellen Sie dann mit diesen Einstellungen einen CloudWatch Alarm.

Nächster Schritt

Schritt 7: Löschen Sie die für dieses Tutorial erstellten AWS Ressourcen

Schritt 7: Löschen Sie die für dieses Tutorial erstellten AWS Ressourcen

Im letzten Schritt von Erste Schritte mit Amazon MSK löschen Sie den MSK-Cluster und den Client-Computer, die Sie für dieses Tutorial erstellt haben.

Um die Ressourcen mit dem zu löschen AWS Management Console

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
2. Wählen Sie den Namen Ihres Clusters aus. Zum Beispiel MSK TutorialCluster.
3. Wählen Sie Actions (Aktionen) und dann Delete (Löschen).
4. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
5. Wählen Sie die Instance aus, die Sie für Ihren Client-Computer erstellt haben, z. B. **`MSKTutorialClient`**.
6. Wählen Sie Instance-Status und dann Instance beenden.

So löschen Sie die IAM-Richtlinie und -Rolle

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Rollen.
3. Geben Sie in das Suchfeld den Namen der IAM-Rolle ein, die Sie für dieses Tutorial erstellt haben.
4. Wählen Sie die Rolle aus. Wählen Sie dann Rolle löschen und bestätigen Sie das Löschen.
5. Wählen Sie im Navigationsbereich Richtlinien.
6. Geben Sie in das Suchfeld den Namen der Richtlinie ein, die Sie für dieses Tutorial erstellt haben.
7. Wählen Sie die Richtlinie aus, um die zugehörige Übersichtsseite zu öffnen. Wählen Sie auf der Übersicht-Seite der Richtlinie die Option Richtlinie löschen.
8. Wählen Sie Löschen aus.

Amazon MSK: Funktionsweise

Ein Amazon-MSK-Cluster ist die primäre Amazon-MSK-Ressource, die Sie in Ihrem Konto erstellen können. In den Themen in diesem Abschnitt wird beschrieben, wie allgemeine Amazon-MSK-Vorgänge ausgeführt werden. Eine Liste aller Vorgänge, die Sie für einen MSK-Cluster ausführen können, finden Sie im Folgenden:

- Die AWS Management Console
- Die API-Referenz für Amazon MSK
- Die Befehlsreferenz für die Amazon-MSK-CLI

Themen

Erstellen eines Amazon-MSK-Clusters

> **Important**
>
> Sie können die VPC für einen Amazon-MSK-Cluster nach dem Erstellen des Clusters nicht mehr ändern.

Bevor Sie einen Amazon-MSK-Cluster erstellen können, benötigen Sie eine Amazon Virtual Private Cloud (VPC) und müssen Subnetze innerhalb dieser VPC einrichten.

Sie benötigen zwei Subnetze in zwei verschiedenen Availability Zones in der Region USA West (Nordkalifornien). Für andere Regionen, in denen Amazon MSK verfügbar ist, können Sie entweder zwei oder drei Subnetze angeben. Die beiden Subnetze müssen sich in verschiedenen Availability Zones befinden. Wenn Sie einen Cluster erstellen, verteilt Amazon MSK die Broker-Knoten gleichmäßig über die von Ihnen angegebenen Subnetze.

Größen der Makler

Wenn Sie einen Amazon MSK-Cluster erstellen, geben Sie die Größe der Broker an, die er haben soll. Amazon MSK unterstützt die folgenden Brokergrößen:

- 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

M7g-Broker verwenden AWS Graviton-Prozessoren (kundenspezifische ARM-basierte Prozessoren, die von Amazon Web Services entwickelt wurden). M7g-Broker bieten im Vergleich zu vergleichbaren M5-Instances ein besseres Preis-Leistungs-Verhältnis. M7g-Broker verbrauchen weniger Strom als vergleichbare M5-Instances.

M7g Graviton-Broker sind in diesen Regionen nicht verfügbar: CDG (Paris), CGK (Jakarta), CPT (Kapstadt), DXB (Dubai), HKG (Hongkong), KIX (Osaka), LHR (London), MEL (Melbourne), MXP (Mailand), OSU (US-Ost), PDT (US-West), TLV (Tel Aviv), YYC (Calgary), ZRH (Zürich).

MSK unterstützt m7G-Broker auf Clustern, auf denen eine der folgenden Kafka-Versionen ausgeführt wird:

- 2.8.2. mehrstufig
- 3.3.2
- 3.4.0
- 3.5.1
- 3.6.0 mit mehrstufigem Speicher
- 3.7.x
- 3.7.x.kraft

M7g- und M5-Broker bieten eine höhere Ausgangsdurchsatzleistung als T3-Broker und werden für Produktionsworkloads empfohlen. M7g- und M5-Broker können auch mehr Partitionen pro Broker haben als T3-Broker. Verwenden Sie M7g- oder M5-Broker, wenn Sie größere produktionstaugliche Workloads ausführen oder eine größere Anzahl von Partitionen benötigen. Weitere Informationen zu den Instance-Größen M7g und M5 finden Sie unter Amazon EC2 General Purpose Instances.

T3-Broker haben die Möglichkeit, CPU-Guthaben zu verwenden, um die Leistung vorübergehend zu steigern. Verwenden Sie T3-Broker für eine kostengünstige Entwicklung, wenn Sie kleine bis mittlere Streaming-Workloads testen oder Streaming-Workloads mit niedrigem Durchsatz haben, bei denen temporäre Spitzen auftreten. Wir empfehlen Ihnen, einen proof-of-concept Test durchzuführen, um festzustellen, ob T3-Broker für die Produktion oder kritische Workloads ausreichend sind. Weitere Informationen zu den Größen von T3-Brokern finden Sie unter Amazon EC2T3-Instances.

Weitere Informationen zur Auswahl der Broker-Größen finden Sie unter. Bewährte Methoden

Erstellen eines Clusters mit dem AWS Management Console

In diesem Prozess wird die allgemeine Aufgabe beschrieben, einen bereitgestellten Cluster mithilfe benutzerdefinierter Erstellungsoptionen zu erstellen. Sie können in der MSK-Konsole andere Optionen auswählen, um einen serverlosen Cluster zu erstellen.

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
2. Wählen Sie Cluster erstellen.
3. Wählen Sie als Methode zur Clustererstellung die Option Benutzerdefiniert aus.
4. Geben Sie einen Clusternamen an, der eindeutig ist und nicht mehr als 64 Zeichen enthält.

5. Wählen Sie als Clustertyp die Option Bereitgestellt aus, sodass Sie die Anzahl der Broker, die Broker-Größe und die Cluster-Speicherkapazität angeben können.
6. Wählen Sie die Apache Kafka-Version aus, die Sie auf den Brokern ausführen möchten. Um einen Vergleich der MSK-Funktionen zu sehen, die von den einzelnen Apache Kafka-Versionen unterstützt werden, wählen Sie Versionskompatibilität anzeigen aus.
7. Abhängig von der ausgewählten Apache Kafka-Version haben Sie möglicherweise die Möglichkeit, den Metadatenmodus des Clusters zu wählen: ZooKeeper oder KraFT.
8. Wählen Sie auf der Grundlage der Rechen-, Arbeitsspeicher- und Speicheranforderungen des Clusters eine Broker-Größe aus, die für den Cluster verwendet werden soll. Weitere Informationen finden Sie unter ???,
9. Wählen Sie die Anzahl der Zonen aus, auf die die Broker verteilt sind.
10. Geben Sie die Anzahl der Broker an, die MSK in jeder Availability Zone erstellen soll. Das Minimum ist ein Broker pro Availability Zone und das Maximum beträgt 30 Broker pro Cluster für ZooKeeper basierte Cluster und 60 Broker pro Cluster für Kraft-basierte Cluster.
11. Wählen Sie die anfängliche Speichermenge aus, über die Ihr Cluster verfügen soll. Sie können die Speicherkapazität nicht verringern, nachdem Sie den Cluster erstellt haben.
12. Abhängig von der ausgewählten Brokergröße (Instanzgröße) können Sie den Durchsatz für bereitgestellten Speicher pro Broker angeben. Um diese Option zu aktivieren, wählen Sie Broker-Größe (Instanzgröße) kafka.m5.4xlarge oder größer für x86 und kafka.m7g.2xlarge oder größer für Graviton-basierte Instances. Siehe ???.
13. Wählen Sie eine Option für den Cluster-Speichermodus, entweder nur EBS-Speicher oder Tiered Storage und EBS-Speicher.
14. Wenn Sie eine benutzerdefinierte Clusterkonfiguration erstellen und verwenden möchten (oder wenn Sie bereits eine Clusterkonfiguration gespeichert haben), wählen Sie eine Konfiguration aus. Andernfalls können Sie den Cluster mit der Amazon MSK-Standard-Cluster-Konfiguration erstellen. Informationen zu Amazon-MSK-Konfigurationen finden Sie unter Konfiguration.
15. Klicken Sie auf Weiter.
16. Wählen Sie für Netzwerkeinstellungen die VPC aus, die Sie für den Cluster verwenden möchten.
17. Geben Sie basierend auf der Anzahl der Zonen, die Sie zuvor ausgewählt haben, die Availability Zones und Subnetze an, in denen Broker bereitstellen werden. Diese Subnetze müssen zu verschiedenen Availability-Zonen gehören.
18. Sie können eine oder mehrere Sicherheitsgruppen auswählen, denen Sie Zugriff auf Ihren Cluster gewähren möchten (z. B. die Sicherheitsgruppen von Client-Computern). Wenn Sie Sicherheitsgruppen angeben, die mit Ihnen gemeinsam genutzt werden, müssen Sie

sicherstellen, dass Sie über die entsprechenden Berechtigungen verfügen. Insbesondere benötigen Sie die `ec2:DescribeSecurityGroups`-Berechtigung. Verbindung zu einem Amazon MSK-Cluster herstellen.

19. Klicken Sie auf Weiter.
20. Wählen Sie die Zugriffskontrollmethoden und Verschlüsselungseinstellungen des Clusters aus, um Daten bei der Übertragung zwischen Clients und Brokern zu verschlüsseln. Weitere Informationen finden Sie unter the section called “Verschlüsselung während der Übertragung”.
21. Wählen Sie den KMS-Schlüssel aus, den Sie für die Verschlüsselung von Daten im Ruhezustand verwenden möchten. Weitere Informationen finden Sie unter the section called “Verschlüsselung im Ruhezustand”.
22. Klicken Sie auf Weiter.
23. Wählen Sie das gewünschte Monitoring und die gewünschten Tags aus. Dies bestimmt den Satz der Metriken, die Sie erhalten. Weitere Informationen finden Sie unter Überwachung eines Clusters. Amazon CloudWatch -, Prometheus -, Broker Log Delivery - oder Cluster-Tags und wählen Sie dann Weiter aus.
24. Überprüfen Sie die Einstellungen für Ihren Cluster. Sie können zurückgehen und Einstellungen ändern, indem Sie Zurück wählen, um zum vorherigen Konsolenbildschirm zurückzukehren, oder Bearbeiten, um bestimmte Clustereinstellungen zu ändern. Wenn die Einstellungen korrekt sind, wählen Sie Cluster erstellen aus.
25. Überprüfen Sie den Cluster-Status auf der Seite Cluster-Zusammenfassung. Der Status ändert sich von Erstellen auf Aktiv, wenn Amazon MSK den Cluster bereitstellt. Wenn der Status Active lautet, können Sie die Verbindung mit dem Cluster herstellen. Weitere Informationen zu Cluster-Status finden Sie unter Cluster-Status.

Erstellen eines Clusters mit dem AWS CLI

1. Kopieren Sie das folgende JSON und speichern Sie es in einer Datei. Benennen Sie die Datei `brokernodegroupinfo.json`. Ersetzen Sie die Subnetz-IDs im JSON durch die Werte, die Ihren Subnetzen entsprechen. Diese Subnetze müssen sich in verschiedenen Availability Zones befinden. Ersetzen Sie *„Security-Group-ID“* durch die ID mindestens einer Sicherheitsgruppe der Client-VPC. Clients, die diesen Sicherheitsgruppen zugeordnet sind, erhalten Zugriff auf den Cluster. Wenn Sie Sicherheitsgruppen angeben, die für Sie freigegeben wurden, müssen Sie sicherstellen, dass Sie über Berechtigungen für diese verfügen. Insbesondere benötigen Sie die `ec2:DescribeSecurityGroups`-Berechtigung. Ein Beispiel

finden Sie unter Amazon EC2: Ermöglicht es, die mit einer bestimmten VPC verknüpften EC2-Sicherheitsgruppen programmgesteuert und in der Konsole zu verwalten. Speichern Sie abschließend die aktualisierte JSON-Datei auf dem Computer, auf dem Sie sie AWS CLI installiert haben.

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

> **Important**
>
> Geben Sie genau zwei Subnetze an, wenn Sie die Region USA West (Nordkalifornien) verwenden. Für andere Regionen, in denen Amazon MSK verfügbar ist, können Sie entweder zwei oder drei Subnetze angeben. Die von Ihnen angegebenen Subnetze müssen sich in verschiedenen Availability Zones befinden. Wenn Sie einen Cluster erstellen, verteilt Amazon MSK die Broker-Knoten gleichmäßig über die von Ihnen angegebenen Subnetze.

2. Führen Sie den folgenden AWS CLI Befehl in dem Verzeichnis aus, in dem Sie die `brokernodegroupinfo.json` Datei gespeichert haben, und ersetzen Sie *„Your-Cluster-Name“ durch einen Namen* Ihrer Wahl. Als *„Überwachungsebene“*, können Sie einen der folgenden drei Werte angeben: `DEFAULT`, `PER_BROKER` oder `PER_TOPIC_PER_BROKER`. Hinweise zu diesen drei verschiedenen Überwachungsbenen finden Sie unter ???. Der Parameter `enhanced-monitoring` ist optional. Ohne weitere Angaben im `create-cluster`-Befehl erhalten Sie die `DEFAULT`-Überwachungsebene.

```
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"
```

Die Ausgabe des Befehls sieht wie das folgende JSON aus:

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

> **Note**
>
> Der `create-cluster`-Befehl gibt möglicherweise einen Fehler zurück, der besagt, dass ein oder mehrere Subnetze nicht unterstützten Availability Zones angehören. In diesem Fall gibt der Fehler an, welche Availability Zones nicht unterstützt werden. Erstellen Sie Subnetze, bei denen die nicht unterstützten Availability Zones nicht verwendet werden, und versuchen Sie es erneut mit dem `create-cluster`-Befehl.

3. Speichern Sie den Wert des `ClusterArn`-Schlüssels, da Sie ihn zum Ausführen anderer Aktionen im Cluster benötigen.
4. Führen Sie den folgenden Befehl aus, um einen Cluster zu überprüfen `STATE`. Der `STATE`-Wert ändert sich von `CREATING` zu `ACTIVE`, wenn Amazon MSK den Cluster bereitstellt. Wenn der Status `ACTIVE` lautet, können Sie die Verbindung mit dem Cluster herstellen. Weitere Informationen zu Cluster-Status finden Sie unter Cluster-Status.

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

Erstellen eines Clusters mit einer benutzerdefinierten Amazon MSK-Konfiguration mithilfe der AWS CLI

Weitere Informationen zu benutzerdefinierten Amazon-MSK-Konfigurationen und deren Erstellung finden Sie unter Konfiguration.

1. Speichern Sie den folgenden JSON in einer Datei und ersetzen Sie *`configuration-arn`* durch den ARN der Konfiguration, die Sie zum Erstellen des Clusters verwenden möchten.

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

```
}
```

2. Führen Sie den `create-cluster`-Befehl aus und weisen Sie mithilfe der `configuration-info`-Option, auf die JSON-Datei, die Sie im vorherigen Schritt gespeichert haben. Im Folgenden wird ein Beispiel gezeigt.

```
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
```

Im Folgenden finden Sie ein Beispiel für eine erfolgreiche Antwort nach der Ausführung dieses Befehls.

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

Erstellen eines Clusters mit der API

Informationen zum Erstellen eines Clusters mithilfe der API finden Sie unter CreateCluster.

Löschen eines Amazon-MSK-Clusters

> **Note**
>
> Wenn Ihr Cluster über eine Auto-Scaling-Richtlinie verfügt, empfehlen wir, dass Sie die Richtlinie entfernen, bevor Sie den Cluster löschen. Weitere Informationen finden Sie unter Auto Scaling.

Löschen eines Clusters mit dem AWS Management Console

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.

2. Wählen Sie den MSK-Cluster, den Sie löschen möchten, indem Sie das Kontrollkästchen daneben aktivieren.
3. Wählen Sie Löschen und bestätigen Sie das Löschen.

Löschen eines Clusters mit dem AWS CLI

Führen Sie den folgenden Befehl aus und *ClusterArn*ersetzen Sie ihn durch den Amazon-Ressourcennamen (ARN), den Sie bei der Erstellung Ihres Clusters erhalten haben. Wenn Ihnen der ARN für Ihren Cluster nicht vorliegt, finden Sie ihn, indem Sie alle Cluster auflisten. Weitere Informationen finden Sie unter the section called “Auflisten von Clustern”.

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

Löschen eines Clusters mithilfe der API

Informationen zum Löschen eines Clusters mithilfe der API finden Sie unter DeleteCluster.

Abrufen der Bootstrap-Broker für einen Amazon-MSK-Cluster

Holen Sie sich die Bootstrap-Broker mit dem AWS Management Console

Der Begriff Bootstrap-Broker bezieht sich auf eine Liste von Brokern, die ein Apache-Kafka-Client als Ausgangspunkt für die Verbindung mit dem Cluster verwenden kann. Diese Liste umfasst nicht unbedingt alle Broker in einem Cluster.

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
2. Die Tabelle führt alle Cluster für die aktuelle Region unter diesem Konto auf. Wählen Sie den Namen eines Clusters aus, um dessen Beschreibung anzuzeigen.
3. Wählen Sie auf der Seite mit der Cluster-Zusammenfassung die Option Client-Informationen anzeigen. Dies zeigt Ihnen die Bootstrap-Broker sowie die ZooKeeper Apache-Verbindungszeichenfolge.

Holen Sie sich die Bootstrap-Broker mit dem AWS CLI

Führen Sie den folgenden Befehl aus und *ClusterArn*ersetzen Sie ihn durch den Amazon-Ressourcennamen (ARN), den Sie bei der Erstellung Ihres Clusters erhalten haben. Wenn Ihnen

der ARN für Ihren Cluster nicht vorliegt, finden Sie ihn, indem Sie alle Cluster auflisten. Weitere Informationen finden Sie unter the section called “Auflisten von Clustern”.

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

Für einen MSK-Cluster, der the section called “IAM-Zugriffssteuerung” verwendet, sieht die Ausgabe dieses Befehls wie das folgende JSON-Beispiel aus.

```
{
    "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"
}
```

Das folgende Beispiel zeigt die Bootstrap-Broker für einen Cluster, für den der öffentliche Zugriff aktiviert ist. Verwenden Sie den `BootstrapBrokerStringPublicSaslIam` für den öffentlichen Zugriff und die `BootstrapBrokerStringSaslIam` Zeichenfolge für den Zugriff von innen 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"
}
```

Die Bootstrap-Broker-Zeichenfolge sollte drei Broker aus den Availability Zones enthalten, in denen der MSK-Cluster bereitgestellt wird (es sei denn, es sind nur zwei Broker verfügbar).

Abrufen der Bootstrap-Broker mithilfe der API

Informationen zu den Bootstrap-Brokern, die die API verwenden, finden Sie unter GetBootstrapBrokers.

Auflisten von Amazon-MSK-Clustern

Auflisten von Clustern mit dem AWS Management Console

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
2. Die Tabelle führt alle Cluster für die aktuelle Region unter diesem Konto auf. Wählen Sie den Namen eines Clusters aus, um dessen Details anzuzeigen.

Cluster auflisten mit dem AWS CLI

Führen Sie den folgenden Befehl aus.

```
aws kafka list-clusters
```

Auflisten von Clustern mithilfe der API

Eine Liste von Clustern, die die API verwenden, finden Sie unter ListClusters.

Verwaltung von Metadaten

Amazon MSK unterstützt Apache ZooKeeper - oder Kraft-Metadatenverwaltungsmodi.

Ab Apache Kafka Version 3.7.x auf Amazon MSK können Sie Cluster erstellen, die den KraFT-Modus anstelle des Modus verwenden. ZooKeeper Kraft-basierte Cluster verlassen sich bei der Verwaltung von Metadaten auf Controller innerhalb von Kafka.

Themen

- ZooKeeper Modus
- KraFT-Modus

ZooKeeper Modus

Apache ZooKeeper ist „ein zentraler Dienst zur Verwaltung von Konfigurationsinformationen, Benennung, Bereitstellung verteilter Synchronisation und Bereitstellung von Gruppendiensten. All diese Arten von Diensten werden in der einen oder anderen Form von verteilten Anwendungen verwendet“, einschließlich Apache Kafka.

Wenn Ihr Cluster den ZooKeeper Modus verwendet, können Sie die folgenden Schritte ausführen, um die ZooKeeper Apache-Verbindungszeichenfolge abzurufen. Wir empfehlen jedoch, dass Sie den verwenden, `BootstrapServerString` um eine Verbindung zu Ihrem Cluster herzustellen und Administratorvorgänge durchzuführen, da das `--zookeeper` Flag in Kafka 2.5 veraltet ist und aus Kafka 3.0 entfernt wurde.

Abrufen der Apache-Verbindungszeichenfolge mithilfe der ZooKeeper AWS Management Console

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
2. Die Tabelle führt alle Cluster für die aktuelle Region unter diesem Konto auf. Wählen Sie den Namen eines Clusters aus, um dessen Beschreibung anzuzeigen.
3. Wählen Sie auf der Seite mit der Cluster-Zusammenfassung die Option Client-Informationen anzeigen. Dies zeigt Ihnen die Bootstrap-Broker sowie die ZooKeeper Apache-Verbindungszeichenfolge.

Abrufen der ZooKeeper Apache-Verbindungszeichenfolge mithilfe der AWS CLI

1. Wenn Sie den Amazon Resourcennamen (ARN) Ihres Clusters nicht kennen, finden Sie ihn, indem Sie alle Cluster in Ihrem Konto auflisten. Weitere Informationen finden Sie unter the section called “Auflisten von Clustern”.
2. Um die ZooKeeper Apache-Verbindungszeichenfolge zusammen mit anderen Informationen zu Ihrem Cluster abzurufen, führen Sie den folgenden Befehl aus und *`ClusterArn`*ersetzen Sie ihn durch den ARN Ihres Clusters.

   ```
   aws kafka describe-cluster --cluster-arn ClusterArn
   ```

   Die Ausgabe dieses `describe-cluster`-Befehls sieht wie das folgende JSON-Beispiel aus.

   ```
   {
       "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"
    }
}
```

Das vorherige JSON-Beispiel zeigt den `ZookeeperConnectString`-Schlüssel in der Ausgabe des `describe-cluster`-Befehls an. Kopieren Sie den Wert, der diesem Schlüssel entspricht, und speichern Sie ihn, für den Fall, dass Sie ein Thema in Ihrem Cluster erstellen müssen.

> **Important**
>
> Ihr Amazon MSK-Cluster muss sich in dem `ACTIVE` Status befinden, in dem Sie die ZooKeeper Apache-Verbindungszeichenfolge abrufen können. Wenn ein Cluster noch den Status „`CREATING`“ aufweist, enthält die Ausgabe des `describe-cluster`-Befehls „`ZookeeperConnectString`“ nicht. Warten Sie in diesem Fall einige Minuten und führen Sie den `describe-cluster` erneut aus, nachdem der Cluster den Status „`ACTIVE`“ erreicht hat.

Abrufen der ZooKeeper Apache-Verbindungszeichenfolge mithilfe der API

Informationen zum Abrufen der ZooKeeper Apache-Verbindungszeichenfolge mithilfe der API finden Sie unter DescribeCluster.

KraFT-Modus

Amazon MSK hat die Unterstützung für KraFT (Apache Kafka Raft) in Kafka Version 3.7.x eingeführt. Die Apache Kafka-Community hat KraFT entwickelt, um Apache ZooKeeper für die Metadatenverwaltung in Apache Kafka-Clustern zu ersetzen. Im KraFT-Modus werden Cluster-Metadaten innerhalb einer Gruppe von Kafka-Controllern, die Teil des Kafka-Clusters sind, und nicht knotenübergreifend verbreitet. ZooKeeper KraFT-Controller sind ohne zusätzliche Kosten für Sie enthalten und erfordern keine zusätzliche Einrichtung oder Verwaltung durch Sie. Weitere Informationen zu KraFT finden Sie unter KIP-500.

Hier sind einige Punkte, die Sie zum KraFT-Modus auf MSK beachten sollten:

- Der KraFT-Modus ist nur für neue Cluster verfügbar. Sie können den Metadatenmodus nicht wechseln, sobald der Cluster erstellt wurde.
- Auf der MSK-Konsole können Sie einen KRAFT-basierten Cluster erstellen, indem Sie Kafka Version 3.7.x auswählen und im Fenster zur Clustererstellung das Kontrollkästchen KraFT aktivieren.
- Um einen Cluster im KraFT-Modus mithilfe der MSK-API `CreateCluster`oder der `CreateClusterV2`MSK-Operationen zu erstellen, sollten Sie als Version verwenden. `3.7.x.kraft` Verwenden Sie `3.7.x` als Version, um einen Cluster im ZooKeeper Modus zu erstellen.
- Die Anzahl der Partitionen pro Broker ist auf KraFT- und ZooKeeper basierten Clustern identisch. Mit KraFT können Sie jedoch mehr Partitionen pro Cluster hosten, indem Sie mehr Broker in einem Cluster bereitstellen.
- Für die Verwendung des Kraft-Modus auf Amazon MSK sind keine API-Änderungen erforderlich. Wenn Ihre Clients die `--zookeeper` Verbindungszeichenfolge jedoch heute noch verwenden, sollten Sie Ihre Clients so aktualisieren, dass sie die `--bootstrap-server` Verbindungszeichenfolge verwenden, um eine Verbindung zu Ihrem Cluster herzustellen. Das `--zookeeper` Flag ist in Apache Kafka Version 2.5 veraltet und wird ab Kafka Version 3.0 entfernt. Wir empfehlen Ihnen daher, aktuelle Apache Kafka-Client-Versionen und die `--bootstrap-server` Verbindungszeichenfolge für alle Verbindungen zu Ihrem Cluster zu verwenden.

- ZooKeeper Der Modus ist weiterhin für alle veröffentlichten Versionen verfügbar, in denen Zookeeper auch von Apache Kafka unterstützt wird. Einzelheiten Unterstützte Apache Kafka-Versionen zum Ende der Unterstützung für Apache Kafka-Versionen und future Updates finden Sie unter.
- Sie sollten überprüfen, ob alle von Ihnen verwendeten Tools in der Lage sind, Kafka Admin-APIs ohne ZooKeeper Verbindungen zu verwenden. Aktuelle Schritte Verwenden von LinkedIn's Cruise Control für Apache Kafka mit Amazon MSK zur Verbindung Ihres Clusters mit Cruise Control finden Sie unter. Cruise Control enthält auch Anweisungen für den Betrieb von Cruise Control ohne ZooKeeper.
- Sie müssen für administrative Aktionen nicht direkt auf die Kraft-Controller Ihres Clusters zugreifen. Wenn Sie jedoch Open Monitoring zur Erfassung von Metriken verwenden, benötigen Sie auch die DNS-Endpunkte Ihrer Controller, um einige Metriken zu Ihrem Cluster zu sammeln, die sich nicht auf Controller beziehen. Sie können diese DNS-Endpunkte über die MSK-Konsole oder mithilfe der API-Operation abrufen. ListNodes Aktualisierte Schritte Offene Überwachung mit Prometheus zur Einrichtung von Open-Monitoring für Kraft-basierte Cluster finden Sie unter.
- Es gibt keine zusätzlichen CloudWatch Metriken, die Sie für Cluster im KraFt-Modus im Vergleich zu Mode-Clustern überwachen ZooKeeper müssen. MSK verwaltet die in Ihren Clustern verwendeten KraFT-Controller.
- Sie können die Verwaltung von ACLs mithilfe von Clustern im KraFt-Modus mithilfe der `--bootstrap-server` Verbindungszeichenfolge fortsetzen. Sie sollten die `--zookeeper` Verbindungszeichenfolge nicht zur Verwaltung von ACLs verwenden. Siehe Apache Kafka ACLs.
- Im KraFT-Modus werden die Metadaten Ihres Clusters auf KraFT-Controllern innerhalb von Kafka und nicht auf externen ZooKeeper Knoten gespeichert. Daher müssen Sie den Zugriff auf Controller-Knoten nicht separat steuern, wie dies bei ZooKeeper Knoten der Fall ist.

Speicherverwaltung

Amazon MSK bietet Features, die Sie bei der Speicherverwaltung auf Ihren MSK-Clustern unterstützen.

Themen

- Gestaffelte Speicherung
- Hochskalieren von Broker-Speicher
- Bereitstellen des Speicherdurchsatzes

Gestaffelte Speicherung

Gestaffelte Speicherung ist eine kostengünstige Speicherstufe für Amazon MSK, die auf praktisch unbegrenzten Speicherplatz skaliert werden kann, sodass Streaming-Datenanwendungen kostengünstig erstellt werden können.

Sie können einen Amazon-MSK-Cluster erstellen, der mit gestaffeltem Speicher konfiguriert ist, der ein ausgewogenes Verhältnis zwischen Leistung und Kosten bietet. Amazon MSK speichert Streaming-Daten auf einer leistungsoptimierten primären Speicherebene, bis die Aufbewahrungsgrenzen für Apache-Kafka-Themen erreicht sind. Anschließend verschiebt Amazon MSK Daten automatisch in die neue kostengünstige Speicherstufe.

Wenn Ihre Anwendung beginnt, Daten aus dem gestaffelten Speicher zu lesen, können Sie mit einer Erhöhung der Leselatenz für die ersten paar Bytes rechnen. Wenn Sie beginnen, die verbleibenden Daten sequentiell aus der kostengünstigen Stufe zu lesen, können Sie mit Latenzen rechnen, die denen der primären Speicherstufe ähneln. Sie müssen keinen Speicher für die kostengünstige gestaffelte Speicherung bereitstellen oder die Infrastruktur verwalten. Sie können beliebig viele Daten speichern und nur für das bezahlen, was Sie tatsächlich nutzen. Dieses Feature ist mit den in KIP-405: Kafka Tiered Storage eingeführten APIs kompatibel.

Im Folgenden sind einige Funktionen der gestaffelten Speicherung aufgeführt:

- Sie können auf praktisch unbegrenzten Speicherplatz skalieren. Sie müssen nicht raten, wie Sie Ihre Apache-Kafka-Infrastruktur skalieren können.
- Sie können Daten in Ihren Apache-Kafka-Themen länger aufbewahren oder Ihren Themenspeicher vergrößern, ohne die Anzahl der Broker erhöhen zu müssen.
- Es bietet einen längeren Sicherheitspuffer, um unerwartete Verzögerungen bei der Verarbeitung zu bewältigen.
- Sie können alte Daten mit Ihrem vorhandenen Stream-Verarbeitungscode und den Kafka-APIs in der exakten Produktionsreihenfolge erneut verarbeiten.
- Partitionen können schneller wieder ausgeglichen werden, da Daten auf sekundärem Speicher nicht zwischen Broker-Festplatten repliziert werden müssen.
- Daten werden zwischen Brokern und dem gestaffelten Speicher innerhalb der VPC bewegt und nicht über das Internet übertragen.
- Ein Client-Computer kann zum Herstellen einer Verbindung zu neuen Clustern mit aktivierter gestaffelter Speicherung den gleichen Prozess wie zum Herstellen einer Verbindung zu einem

Cluster ohne aktivierte gestaffelte Speicherung verwenden. Siehe Erstellen eines Client-Computers.

Anforderungen für gestaffelte Speicherung

- Sie müssen den Apache-Kafka-Client Version 3.0.0 oder höher verwenden, um ein neues Thema mit aktivierter gestaffelter Speicherung zu erstellen. Um ein vorhandenes Thema auf gestaffelte Speicherung umzustellen, können Sie einen Client-Computer neu konfigurieren, der eine Kafka-Client-Version unter 3.0.0 verwendet (die unterstützte Apache-Kafka-Version ist mindestens 2.8.2.tiered), um die gestaffelte Speicherung zu aktivieren. Siehe Schritt 4: Ein Thema erstellen.
- Der Amazon MSK-Cluster mit aktiviertem Tiered Storage muss Version 3.6.0 oder höher oder 2.8.2. Tiered verwenden.

Einschränkungen und Limits bei der gestaffelten Speicherung

Für die gestaffelte Speicherung gelten die folgenden Einschränkungen und Limits:

- Die gestaffelte Speicherung gilt nur für Cluster im Bereitstellungsmodus.
- Tiered Storage unterstützt die Brokergröße t3.small nicht.
- Die Mindestaufbewahrungsdauer bei kostengünstiger Speicherung beträgt 3 Tage. Es gibt keine Mindestaufbewahrungsdauer für den Primärspeicher.
- Die gestaffelte Speicherung unterstützt nicht mehrere Protokollverzeichnisse auf einem Broker (JBOD-bezogene Funktionen).
- Die gestaffelte Speicherung unterstützt keine komprimierten Themen. Stellen Sie sicher, dass für alle Themen, für die die gestaffelte Speicherung aktiviert ist, die cleanup.policy nur auf „DELETE“ konfiguriert ist.
- Die gestaffelte Speicherung kann für einzelne Themen deaktiviert werden, jedoch nicht für den gesamten Cluster. Nach der Deaktivierung kann die gestaffelte Speicherung für ein Thema nicht wieder aktiviert werden.
- Wenn Sie Amazon MSK Version 2.8.2.tiered verwenden, können Sie nur zu einer anderen von Tiered Storage unterstützten Apache Kafka-Version migrieren. Wenn Sie eine von Tiered Storage unterstützte Version nicht weiter verwenden möchten, erstellen Sie einen neuen MSK-Cluster und migrieren Sie Ihre Daten dorthin.
- Das kafka-log-dirs Tool kann die Größe von Tiered Storage-Daten nicht melden. Das Tool meldet nur die Größe der Protokollsegmente im Primärspeicher.

Wie Protokollsegmente in den gestaffelten Speicher kopiert werden

Wenn Sie gestaffelte Speicherung für ein neues oder vorhandenes Thema aktivieren, kopiert Apache Kafka geschlossene Protokollsegmente vom Primärspeicher in den gestaffelten Speicher.

- Apache Kafka kopiert nur geschlossene Protokollsegmente. Es kopiert alle Nachrichten innerhalb des Protokollsegments in einen gestaffelten Speicher.
- Aktive Segmente kommen nicht für gestaffelte Speicherung in Frage. Die Größe des Protokollsegments (segment.bytes) oder die Segment-Rollzeit (segment.ms) steuern die Geschwindigkeit, mit der Segmente geschlossen werden, und die Geschwindigkeit, mit der Apache Kafka sie anschließend in den gestaffelten Speicher kopiert.

Die Aufbewahrungseinstellungen für ein Thema mit aktivierter gestaffelter Speicherung unterscheiden sich von den Einstellungen für ein Thema ohne aktivierte gestaffelte Speicherung. Die folgenden Regeln steuern die Aufbewahrung von Nachrichten in Themen, für die gestaffelte Speicherung aktiviert ist:

- Sie definieren die Aufbewahrung in Apache Kafka mit zwei Einstellungen: log.retention.ms (Zeit) und log.retention.bytes (Größe). Diese Einstellungen bestimmen die Gesamtdauer und Größe der Daten, die Apache Kafka im Cluster aufbewahrt. Unabhängig davon, ob Sie den gestaffelten Speichermodus aktivieren oder nicht, legen Sie diese Konfigurationen auf Cluster-Ebene fest. Sie können die Einstellungen auf Themenebene mit Themenkonfigurationen überschreiben.
- Wenn Sie die gestaffelte Speicherung aktivieren, können Sie zusätzlich angeben, wie lange die primäre Hochleistungs-Speicherebene Daten speichert. Wenn für ein Thema beispielsweise die Einstellung für die gesamte Aufbewahrung (log.retention.ms) von 7 Tagen und die lokale Aufbewahrung (local.retention.ms) für 12 Stunden festgelegt ist, speichert der primäre Speicher des Clusters Daten nur für die ersten 12 Stunden. Bei der kostengünstigen Speicherstufe werden die Daten für die gesamten 7 Tage aufbewahrt.
- Die üblichen Aufbewahrungseinstellungen gelten für das gesamte Protokoll. Dazu gehören auch die gestaffelten und die primären Komponenten.
- Die Einstellungen local.retention.ms oder local.retention.bytes steuern die Aufbewahrung von Nachrichten im Primärspeicher. Wenn Daten in einem vollständigen Protokoll die Schwellenwerte für die Aufbewahrung im Primärspeicher (local.retention.ms/bytes) erreicht haben, kopiert Apache Kafka die Daten im Primärspeicher in den gestaffelten Speicher. Die Daten können dann ablaufen.

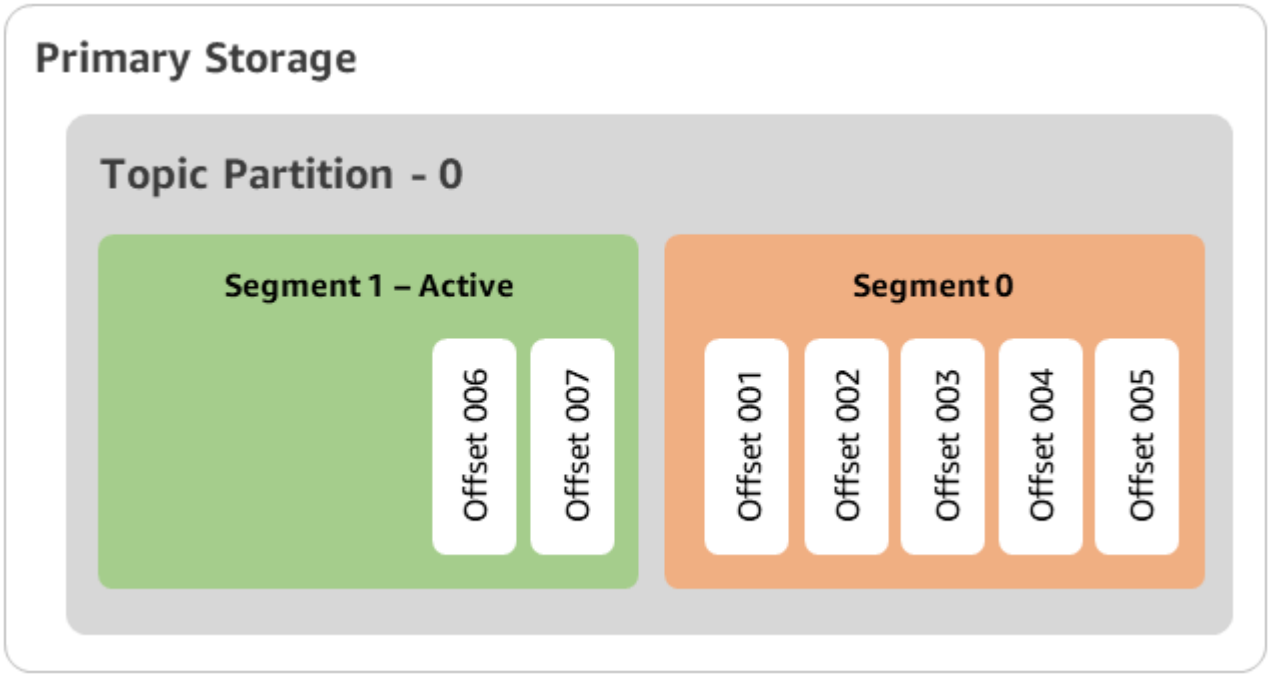
- Wenn Apache Kafka eine Nachricht in einem Protokollsegment in einen gestaffelten Speicher kopiert, entfernt es die Nachricht auf der Grundlage der Einstellungen retention.ms oder retention.bytes aus dem Cluster.

Beispielsszenario mit gestaffelter Speicherung

Dieses Szenario veranschaulicht, wie sich ein vorhandenes Thema, das Nachrichten im Primärspeicher enthält, verhält, wenn gestaffelte Speicherung aktiviert ist. Sie aktivieren die gestaffelte Speicherung zu diesem Thema, indem Sie remote.storage.enable auf `true` setzen. In diesem Beispiel ist retention.ms auf 5 Tage und local.retention.ms auf 2 Tage festgelegt. Im Folgenden ist die Reihenfolge der Ereignisse dargestellt, wenn ein Segment abläuft.

Zeitpunkt T0 - Bevor Sie die gestaffelte Speicherung aktivieren.

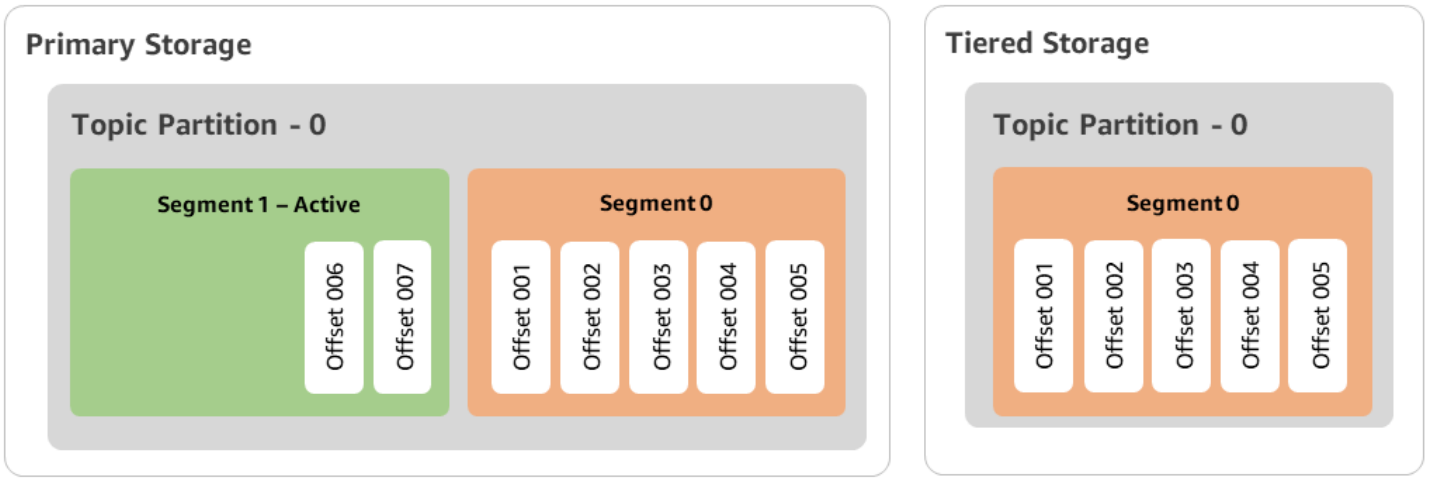
Bevor Sie die gestaffelte Speicherung für dieses Thema aktivieren, gibt es zwei Protokollsegmente. Eines der Segmente ist für eine bestehende Themenpartition 0 aktiv.



Zeitpunkt T1 (< 2 Tage) - gestaffelte Speicherung aktiviert. Segment 0 wurde in den gestaffelten Speicher kopiert.

Nachdem Sie die gestaffelte Speicherung für dieses Thema aktiviert haben, kopiert Apache Kafka das Protokollsegment 0 in den gestaffelten Speicher, nachdem das Segment die ursprünglichen Aufbewahrungseinstellungen erreicht hat. Apache Kafka behält auch die primäre Speicherkopie von

Segment 0 bei. Das aktive Segment 1 ist noch nicht berechtigt, auf den gestaffelten Speicher zu kopieren. In diesem Zeitplan wendet Amazon MSK noch keine der Aufbewahrungseinstellungen für Nachrichten in Segment 0 und Segment 1 an. (local.retention.bytes/ms, retention.ms/bytes)



Zeitpunkt T2 – Die lokale Aufbewahrung ist wirksam.

Nach 2 Tagen werden die primären Aufbewahrungseinstellungen für das Segment 0 wirksam, das Apache Kafka in den gestaffelten Speicher kopiert hat. Dies wird durch die Einstellung von local.retention.ms auf 2 Tage festgelegt. Segment 0 läuft jetzt im Primärspeicher ab. Das aktive Segment 1 ist noch nicht ablauffähig und kann auch nicht in den gestaffelten Speicher kopiert werden.

Zeitpunkt T3 - Die gesamte Aufbewahrung ist wirksam.

Nach 5 Tagen werden die Aufbewahrungseinstellungen wirksam, und Kafka löscht das Protokollsegment 0 und die zugehörigen Nachrichten aus dem gestaffelten Speicher. Segment 1 ist

noch nicht ablauffähig und kann auch nicht in den gestaffelten Speicher kopiert werden, da es noch aktiv ist. Segment 1 ist noch nicht geschlossen und kommt daher nicht für Segment-Rolling in Frage.

Erstellen eines Amazon MSK-Clusters mit mehrstufigem Speicher mit dem AWS Management Console

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
2. Wählen Sie Cluster erstellen.
3. Wählen Sie Benutzerdefiniert erstellen für die gestaffelte Speicherung.
4. Geben Sie einen Namen für den Cluster ein.
5. Wählen Sie unter Cluster-Typ die Option Bereitgestellt aus.
6. Wählen Sie eine Amazon-Kafka-Version aus, welche die gestaffelte Speicherung für Amazon MSK zur Erstellung des Clusters unterstützt.
7. Geben Sie eine andere Broker-Größe als kafka.t3.small an.
8. Geben Sie die Anzahl der Broker an, die Amazon MSK in jeder Availability Zone erstellen soll. Mindestens ist ein Broker pro Availability Zone erforderlich und maximal sind 30 Broker pro Cluster möglich.
9. Geben Sie die Anzahl der Zonen an, auf die Broker verteilt sind.
10. Geben Sie die Anzahl der Apache-Kafka-Broker an, die pro Zone bereitgestellt werden.
11. Wählen Sie Speicheroptionen. Dazu gehören Tiered Storage und EBS Storage zur Aktivierung des gestaffelten Speichermodus.
12. Führen Sie die restlichen Schritte im Cluster-Erstellungsassistenten aus. Wenn der Vorgang abgeschlossen ist, werden Tiered Storage und EBS Storage in der Ansicht Überprüfen und Erstellen als Cluster-Speichermodus angezeigt.

13. Wählen Sie Cluster erstellen aus.

Erstellen eines Amazon MSK-Clusters mit mehrstufigem Speicher mit dem AWS CLI

Um die gestaffelte Speicherung auf einem Cluster zu aktivieren, erstellen Sie den Cluster mit der richtigen Apache-Kafka-Version und dem richtigen Attribut für gestaffelte Speicherung. Folgen Sie dem folgenden Codebeispiel. Führen Sie außerdem die im nächsten Abschnitt beschriebenen Schritte aus, um Erstellen eines Kafka-Themas mit aktiviertem gestaffelten Speicher.

Eine vollständige Liste der unterstützten Attribute für die Cluster-Erstellung finden Sie unter create-cluster.

```
aws tiered-storage 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"
```

Erstellen eines Kafka-Themas mit aktiviertem gestaffelten Speicher

Um den Prozess abzuschließen, den Sie bei der Erstellung eines Clusters mit aktivierter gestaffelter Speicherung gestartet haben, erstellen Sie auch ein Thema mit aktivierter gestaffelter Speicherung mit den Attributen aus dem späteren Codebeispiel. Die spezifischen Attribute für die gestaffelte Speicherung lauten wie folgt:

- `local.retention.ms` (z. B. 10 Minuten) für zeitbasierte Aufbewahrungseinstellungen oder `local.retention.bytes` für Größenbeschränkungen für Protokollsegmente.
- `remote.storage.enable` auf `true` gesetzt, um die gestaffelte Speicherung zu aktivieren.

Die folgende Konfiguration verwendet local.retention.ms, aber Sie können dieses Attribut durch local.retention.bytes ersetzen. Dieses Attribut steuert die Zeit, die vergehen kann, oder die Anzahl der Byte, die Apache Kafka kopieren kann, bevor Apache Kafka die Daten vom Primärspeicher in den gestaffelten Speicher kopiert. Weitere Informationen zu den unterstützten Konfigurationsattributen finden Sie unter Konfiguration auf Themenebene.

> **Note**
>
> Sie müssen den Apache-Kafka-Client Version 3.0.0 und höher verwenden. Diese Versionen unterstützen die Einstellung `remote.storage.enable` nur in diesen Client-Versionen von `kafka-topics.sh`. Informationen zur Aktivierung der gestaffelten Speicherung für ein vorhandenes Thema, das eine frühere Version von Apache Kafka verwendet, finden Sie im Abschnitt Aktivieren der gestaffelten Speicherung für ein vorhandenes Thema.

```
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
```

Aktivieren und Deaktivieren der gestaffelten Speicherung bei einem vorhandenen Thema

In diesen Abschnitten wird beschrieben, wie Sie die gestaffelte Speicherung für ein Thema aktivieren und deaktivieren, das Sie bereits erstellt haben. Informationen zum Erstellen eines neuen Clusters und Themas mit aktiviertem gestaffelten Speicher finden Sie unter Erstellen eines Clusters mit gestaffeltem Speicher mithilfe der AWS Management Console.

Aktivieren der gestaffelten Speicherung für ein vorhandenes Thema

Verwenden Sie die `alter`-Befehlssyntax im folgenden Beispiel, um die gestaffelte Speicherung für ein vorhandenes Thema zu aktivieren. Wenn Sie die gestaffelte Speicherung für ein bereits vorhandenes Thema aktivieren, sind Sie nicht auf eine bestimmte Apache-Kafka-Client-Version beschränkt.

```
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'
```

Deaktivieren der gestaffelten Speicherung für ein vorhandenes Thema

Um die gestaffelte Speicherung für ein vorhandenes Thema zu deaktivieren, verwenden Sie die `alter`-Befehlssyntax in derselben Reihenfolge wie bei der Aktivierung der gestaffelten Speicherung.

```
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**
>
> Wenn Sie die gestaffelte Speicherung deaktivieren, löschen Sie die Themendaten in der gestaffelten Speicherung vollständig. Apache Kafka behält primäre Speicherdaten bei, wendet aber weiterhin die primären Aufbewahrungsregeln anhand von `local.retention.ms` an. Wenn Sie die gestaffelte Speicherung für ein Thema deaktiviert haben, können Sie sie nicht erneut aktivieren. Wenn Sie die gestaffelte Speicherung für ein bereits vorhandenes Thema deaktivieren, sind Sie nicht auf eine bestimmte Apache-Kafka-Client-Version beschränkt.

Tiered Storage auf einem vorhandenen Cluster mithilfe von AWS CLI aktivieren

> **Note**
>
> Sie können die gestaffelte Speicherung nur aktivieren, wenn die log.cleanup.policy Ihres Clusters auf `delete` eingestellt ist, da komprimierte Themen bei der gestaffelten Speicherung nicht unterstützt werden. Später können Sie die log.cleanup.policy eines einzelnen Themas auf `compact` konfigurieren, wenn die gestaffelte Speicherung für dieses bestimmte Thema nicht aktiviert ist. Weitere Informationen zu den unterstützten Konfigurationsattributen finden Sie unter Konfiguration auf Themenebene.

1. Die Kafka-Version aktualisieren – Cluster-Versionen sind keine einfachen Ganzzahlen. Verwenden Sie den Befehl `DescribeCluster` operation oder den `describe-cluster` AWS CLI-Befehl, um die aktuelle Version des Clusters zu ermitteln. `KTVPDKIKX0DER` ist ein Beispiel für eine Version.

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

2. Den Cluster-Speichermodus bearbeiten. Das folgende Codebeispiel zeigt die Bearbeitung des Cluster-Speichermodus auf `TIERED` mithilfe der `update-storage`-API.

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

Aktualisieren des gestaffelten Speichers auf einem vorhandenen Cluster mithilfe der Konsole

> **Note**
>
> Sie können die gestaffelte Speicherung nur aktivieren, wenn die log.cleanup.policy Ihres Clusters auf `delete` eingestellt ist, da komprimierte Themen bei der gestaffelten Speicherung nicht unterstützt werden. Später können Sie die log.cleanup.policy eines einzelnen Themas auf `compact` konfigurieren, wenn die gestaffelte Speicherung für dieses bestimmte Thema nicht aktiviert ist. Weitere Informationen zu den unterstützten Konfigurationsattributen finden Sie unter Konfiguration auf Themenebene.

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
2. Rufen Sie die Cluster-Übersichtsseite auf und wählen Sie Eigenschaften.
3. Rufen Sie den Bereich Speicher auf und wählen Sie Cluster-Speichermodus bearbeiten.
4. Wählen Sie Gestaffelter Speicher und EBS-Speicher und Änderungen speichern.

Hochskalieren von Broker-Speicher

Sie können die Menge an EBS-Speicher pro Broker erhöhen. Sie können den Speicher nicht verringern.

Während dieses Skalierungsvorgangs bleiben Speichervolumen verfügbar.

> **Important**
>
> Wenn der Speicher für einen MSK-Cluster skaliert wird, wird der zusätzliche Speicher sofort verfügbar gemacht. Der Cluster benötigt jedoch nach jedem Speicher-Skalierungsereignis eine Abkühlphase. Amazon MSK verwendet diese Abkühlphase, um den Cluster zu optimieren, bevor er erneut skaliert werden kann. Dieser Zeitraum kann je nach Speichergröße und Auslastung des Clusters sowie vom Datenverkehr zwischen mindestens

6 Stunden und mehr als 24 Stunden liegen. Dies gilt sowohl für auto Skalierungsereignisse als auch für manuelle Skalierung mithilfe des UpdateBrokerSpeichervorgangs. Informationen zur richtigen Größe Ihres Speichers finden Sie unter Bewährte Methoden.

Sie können gestaffelten Speicher verwenden, um Ihren Broker auf unbegrenzte Speichermengen hochzuskalieren. Siehe Gestaffelte Speicherung.

Themen

- Auto Scaling
- Manuelle Skalierung

Auto Scaling

Um den Speicher Ihres Clusters als Reaktion auf eine erhöhte Auslastung automatisch zu erweitern, können Sie eine Richtlinie zur automatischen Skalierung von Anwendungen für Amazon MSK konfigurieren. In einer Auto-Scaling-Richtlinie legen Sie die Ziel-Festplattenauslastung und die maximale Skalierungskapazität fest.

Bevor Sie die automatische Skalierung für Amazon MSK verwenden, sollten Sie Folgendes berücksichtigen:

- **Important**

  Eine Speicher-Skalierungsaktion kann nur einmal alle sechs Stunden ausgeführt werden.

  Wir empfehlen, dass Sie mit einem Speichervolumen beginnen, das Ihren Speicheranforderungen entspricht. Hinweise zur Dimensionierung Ihrer MSK-Cluster finden Sie unter Die Größe Ihres Clusters anpassen: Anzahl der Broker pro Cluster.
- Amazon MSK reduziert den Cluster-Speicher nicht als Reaktion auf eine geringere Nutzung. Amazon MSK unterstützt die Verringerung der Größe von Speichervolumes nicht. Wenn Sie die Größe Ihres Cluster-Speichers reduzieren müssen, müssen Sie Ihren vorhandenen Cluster auf einen Cluster mit kleinerem Speicher migrieren. Weitere Informationen zur Migration eines Clusters finden Sie unter Migration.
- Amazon MSK unterstützt die automatische Skalierung in den Regionen Asien-Pazifik (Osaka) und Afrika (Kapstadt) nicht.

- Wenn Sie Ihrem Cluster eine Auto-Scaling-Richtlinie zuordnen, erstellt Amazon EC2 Auto Scaling automatisch einen CloudWatch Amazon-Alarm für die Zielverfolgung. Wenn Sie einen Cluster mit einer Auto-Scaling-Richtlinie löschen, bleibt dieser CloudWatch Alarm bestehen. Um den CloudWatch Alarm zu löschen, sollten Sie eine Auto-Scaling-Richtlinie aus einem Cluster entfernen, bevor Sie den Cluster löschen. Informationen zur Ziel-Nachverfolgung finden Sie unter Skalierungsrichtlinien für die Ziel-Nachverfolgung für Amazon EC2 Auto Scaling im Benutzerhandbuch für Amazon EC2 Auto Scaling.

Einzelheiten zur Auto-Scaling-Richtlinie

Eine Auto-Scaling-Richtlinie definiert die folgenden Parameter für Ihren Cluster:

- Speichernutzungsziel: Der Schwellenwert für die Speichernutzung, den Amazon MSK zum Auslösen eines Auto-Scaling-Vorgangs verwendet. Sie können das Nutzungsziel auf zwischen 10 % und 80 % der aktuellen Speicherkapazität festlegen. Wir empfehlen, das Speichernutzungsziel auf zwischen 50 % und 60 % festzulegen.
- Maximale Speicherkapazität: Die maximale Skalierungsgrenze, die Amazon MSK für Ihren Broker-Speicher festlegen kann. Sie können die maximale Speicherkapazität auf bis zu 16 TiB pro Broker festlegen. Weitere Informationen finden Sie unter Amazon-MSK-Kontingent.

Wenn Amazon MSK feststellt, dass Ihre `Maximum Disk Utilization`-Metrik gleich oder größer als die `Storage Utilization Target`-Einstellung ist, erhöht es Ihre Speicherkapazität um eine Menge, die der größeren von zwei Zahlen entspricht: 10 GiB oder 10 % des aktuellen Speichers. Wenn Sie beispielsweise 1000 GiB haben, ist diese Menge 100 GiB. Der Service überprüft die Speichernutzung jede Minute. Durch weitere Skalierungsvorgänge wird der Speicherplatz weiter um eine Menge erhöht, die der größeren von zwei Zahlen entspricht: 10 GiB oder 10 % des aktuellen Speichers.

Verwenden Sie den ListClusterOperationsVorgang, um festzustellen, ob auto-scaling Skalierungsvorgänge stattgefunden haben.

Einrichtung der automatischen Skalierung für Ihren Amazon-MSK-Cluster

Sie können die Amazon MSK-Konsole, die Amazon MSK-API oder die automatische Skalierung für AWS CloudFormation den Speicher verwenden. CloudFormation Support ist verfügbar über. Application Auto Scaling

> **Note**
>
> Es ist nicht möglich, eine automatische Skalierung festzulegen, wenn Sie einen Cluster erstellen. Sie müssen zuerst den Cluster erstellen und dann eine Auto-Scaling-Richtlinie für diesen erstellen und aktivieren. Sie können die Richtlinie jedoch erstellen, während der Amazon-MSK-Service Ihren Cluster erstellt.

Einrichtung der automatischen Skalierung mit der AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon MSK-Konsole unter https://console.aws.amazon.com/msk/home?region=us-east-1#/home/.
2. Wählen Sie Ihren Cluster in der Liste der Cluster aus. Dadurch gelangen Sie zu einer Seite, auf der Details zum Cluster aufgeführt sind.
3. Wählen Sie im Abschnitt Auto Scaling für Speicher die Option Konfigurieren.
4. Erstellen und benennen Sie eine Auto-Scaling-Richtlinie. Geben Sie das Speichernutzungsziel, die maximale Speicherkapazität und die Zielmetrik an.
5. Wählen Sie `Save changes`.

Wenn Sie die neue Richtlinie speichern und aktivieren, wird die Richtlinie für den Cluster aktiv. Amazon MSK erweitert dann den Speicher des Clusters, wenn das Speichernutzungsziel erreicht ist.

Einrichtung der automatischen Skalierung mit der CLI

1. Verwenden Sie den RegisterScalableTargetBefehl, um ein Speichernutzungsziel zu registrieren.
2. Verwenden Sie den PutScalingPolicyBefehl, um eine automatische Erweiterungsrichtlinie zu erstellen.

Einrichtung der automatischen Skalierung mit der API

1. Verwenden Sie die RegisterScalableTargetAPI, um ein Speichernutzungsziel zu registrieren.
2. Verwenden Sie die PutScalingPolicyAPI, um eine automatische Erweiterungsrichtlinie zu erstellen.

Manuelle Skalierung

Warten Sie mit der Speichererhöhung, bis sich der Cluster im Status `ACTIVE` befindet. Bei der Speicherskalierung gibt es zwischen Ereignissen eine Abkühlzeit von mindestens sechs Stunden. Der Vorgang stellt zwar sofort zusätzlichen Speicher zur Verfügung, der Service führt jedoch Optimierungen an Ihrem Cluster durch, die bis zu 24 Stunden oder länger dauern können. Die Dauer dieser Optimierungen ist proportional zur Speichergröße.

Skalierung des Broker-Speichers mit dem AWS Management Console

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
2. Wählen Sie den MSK-Cluster aus, für den Sie Broker-Speicher aktualisieren möchten.
3. Wählen Sie im Abschnitt Speicher die Option Bearbeiten aus.
4. Geben Sie das gewünschte Speicher-Volume an. Sie können die Speichermenge nur erhöhen, nicht verringern.
5. Wählen Sie Änderungen speichern aus.

Skalierung des Broker-Speichers mit dem AWS CLI

Führen Sie den folgenden Befehl aus und *`ClusterArn`*ersetzen Sie ihn durch den Amazon-Ressourcennamen (ARN), den Sie bei der Erstellung Ihres Clusters erhalten haben. Wenn Ihnen der ARN für Ihren Cluster nicht vorliegt, finden Sie ihn, indem Sie alle Cluster auflisten. Weitere Informationen finden Sie unter the section called “Auflisten von Clustern”.

Ersetzen Sie *`Aktuelle-Cluster-Version`* durch die aktuelle Version des Clusters.

> **Important**
>
> Cluster-Versionen sind keine einfachen Ganzzahlen. Um die aktuelle Version des Clusters zu finden, verwenden Sie den DescribeClusterBefehl operation oder describe-cluster AWS CLI . `KTVPDKIKX0DER` ist ein Beispiel für eine Version.

Der Parameter *`Target-Volume-in-GiB`* stellt die Speichermenge dar, die jeder Broker haben soll. Es ist nur möglich, den Speicher für alle Broker zu aktualisieren. Sie können keine einzelnen Broker angeben, für die der Speicher aktualisiert werden soll. Der Wert, den Sie bei *`Target-Volume-in-GiB`* angeben, muss eine ganze Zahl sein, die größer als 100 GiB ist. Der Speicher pro Broker darf nach dem Aktualisierungsvorgang den Wert 16384 GiB nicht überschreiten.

```
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}'
```

Hochskalieren von Broker-Speicher mithilfe der API

Informationen zum Aktualisieren eines Broker-Speichers mithilfe der API finden Sie unter UpdateBroker Speicher.

Bereitstellen des Speicherdurchsatzes

Amazon-MSK-Broker speichern Daten auf Speichervolumes. Speicher-I/O wird verbraucht, wenn Produzenten in den Cluster schreiben, wenn Daten zwischen Brokern repliziert werden und wenn Verbraucher Daten lesen, die sich nicht im Arbeitsspeicher befinden. Der Volumenspeicherdurchsatz ist die Geschwindigkeit, mit der Daten in ein Speichervolume geschrieben und von diesem gelesen werden können. Beim bereitgestellten Speicherdurchsatz handelt es sich um die Möglichkeit, diese Rate für die Broker in Ihrem Cluster festzulegen.

Sie können die bereitgestellte Durchsatzrate in MiB pro Sekunde für Cluster angeben, deren Broker größer `kafka.m5.4xlarge` oder größer sind und wenn das Speichervolumen 10 GiB oder mehr beträgt. Es ist möglich, den bereitgestellten Durchsatz bei der Cluster-Erstellung anzugeben. Sie können den bereitgestellten Durchsatz auch für einen Cluster aktivieren oder deaktivieren, der sich im Status `ACTIVE` befindet.

Durchsatz-Engpässe

Es gibt mehrere Ursachen für Engpässe beim Broker-Durchsatz: den Volumendurchsatz, den Netzwerkdurchsatz von Amazon EC2 zu Amazon EBS und den Amazon-EC2-Ausgangsdurchsatz. Sie können den bereitgestellten Speicherdurchsatz aktivieren, um den Volumendurchsatz anzupassen. Einschränkungen des Broker-Durchsatzes können jedoch durch den Netzwerkdurchsatz von Amazon EC2 zu Amazon EBS und den Amazon-EC2-Ausgangsdurchsatz verursacht werden.

Der Amazon-EC2-Ausgangsdurchsatz wird von der Anzahl der Verbrauchergruppen und der Verbraucher pro Verbrauchergruppe beeinflusst. Außerdem sind sowohl der Netzwerkdurchsatz von Amazon EC2 zu Amazon EBS als auch der Amazon EC2 EC2-Ausgangsdurchsatz bei größeren Brokern höher.

Für Volumengrößen von 10 GiB oder mehr können Sie einen Speicherdurchsatz von 250 MiB pro Sekunde oder mehr bereitstellen. 250 MiB pro Sekunde ist die Standardeinstellung. Um den

Speicherdurchsatz bereitzustellen, müssen Sie die Broker-Größe kafka.m5.4xlarge oder größer (oder kafka.m7g.2xlarge oder größer) wählen, und Sie können den maximalen Durchsatz angeben, wie in der folgenden Tabelle dargestellt.

Größe des Brokers	Maximaler Speicherdurchsatz (MiB/s)
kafka.m5.4xlarge	593
kafka.m5.8xlarge	850
kafka.m5.12xlarge	1000
kafka.m 5.16x groß	1000
kafka.m5.24xlarge	1000
kafka.m 7 g, 2 x groß	312,5
kafka.m7g.4x groß	625
kafka.m7g.8xgroß	1000
kafka.m7g.12x groß	1000
kafka.m7g.16x groß	1000

Messung des Speicherdurchsatzes

Sie können die Metriken `VolumeReadBytes` und `VolumeWriteBytes` verwenden, um den durchschnittlichen Speicherdurchsatz eines Clusters zu messen. Die Summe dieser beiden Metriken ergibt den durchschnittlichen Speicherdurchsatz in Bytes. Um den durchschnittlichen Speicherdurchsatz für einen Cluster zu ermitteln, setzen Sie diese beiden Metriken auf SUM und den Zeitraum auf 1 Minute, und verwenden Sie dann die folgende Formel.

```
Average storage throughput in MiB/s = (Sum(VolumeReadBytes) + Sum(VolumeWriteBytes)) /
 (60 * 1024 * 1024)
```

Weitere Informationen über die Metriken `VolumeReadBytes` und `VolumeWriteBytes` finden Sie unter the section called “Überwachung auf PER_BROKER-Ebene”.

Aktualisierung der Konfiguration

Sie können Ihre Amazon-MSK-Konfiguration entweder vor oder nach der Aktivierung des bereitgestellten Durchsatzes aktualisieren. Der gewünschte Durchsatz wird Ihnen jedoch erst angezeigt, wenn Sie beide Aktionen ausführen: den Konfigurationsparameter `num.replica.fetchers` aktualisieren und den bereitgestellten Durchsatz aktivieren.

In der Standardkonfiguration von Amazon MSK hat `num.replica.fetchers` den Wert 2. Sie können Ihr `num.replica.fetchers` aktualisieren, indem Sie die vorgeschlagenen Werte aus der folgenden Tabelle verwenden. Diese Werte dienen zur Orientierung. Wir empfehlen Ihnen, diese Werte an Ihren Anwendungsfall anzupassen.

Größe des Maklers	num.replica.fetchers
kafka.m5.4xlarge	4
kafka.m5.8xlarge	8
kafka.m5.12xlarge	14
kafka.m 5.16x groß	16
kafka.m5.24xlarge	16

Ihre aktualisierte Konfiguration wird möglicherweise erst nach 24 Stunden wirksam und kann länger dauern, wenn ein Quell-Volume nicht voll ausgelastet ist. Die Leistung eines temporären Volumes entspricht jedoch mindestens der Leistung der Quell-Speicher-Volumes während des Migrationszeitraums. Die Migration eines voll ausgelasteten 1-TiB-Volumes zu einer aktualisierten Konfiguration dauert in der Regel etwa sechs Stunden.

Bereitstellung des Speicherdurchsatzes mithilfe der AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon MSK-Konsole unter https://console.aws.amazon.com/msk/home?region=us-east-1#/home/.
2. Wählen Sie Cluster erstellen.
3. Wählen Sie Benutzerdefiniert erstellen.
4. Geben Sie einen Namen für den Cluster ein.
5. Wählen Sie im Abschnitt Speicher die Option Aktivieren.

6. Wählen Sie einen Wert für den Speicherdurchsatz pro Broker.
7. Wählen Sie eine VPC, Zonen und Subnetze und eine Sicherheitsgruppe.
8. Wählen Sie Weiter aus.
9. Wählen Sie unten im Schritt Sicherheit die Option Weiter.
10. Wählen Sie unten im Schritt Überwachung und Tags die Option Weiter.
11. Überprüfen Sie die Cluster-Einstellungen und wählen Sie dann Cluster erstellen.

Bereitstellung des Speicherdurchsatzes mithilfe des AWS CLI

Dieser Abschnitt zeigt ein Beispiel dafür, wie Sie den verwenden können AWS CLI , um einen Cluster mit aktiviertem bereitgestellten Durchsatz zu erstellen.

1. Kopieren Sie den folgenden JSON-Code in eine Datei. Ersetzen Sie die Platzhalter für Subnetz- und Sicherheitsgruppen-IDs durch Ihre eigenen Werte. Benennen Sie die Datei `cluster-creation.json` und speichern Sie sie.

```
{
    "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. Führen Sie den folgenden AWS CLI Befehl in dem Verzeichnis aus, in dem Sie die JSON-Datei im vorherigen Schritt gespeichert haben.

```
aws kafka create-cluster-v2 --cli-input-json file://cluster-creation.json
```

Bereitstellen des Speicherdurchsatzes mit der API

Verwenden Sie CreateClusterV2, um den bereitgestellten Speicherdurchsatz bei der Erstellung eines Clusters zu konfigurieren.

Aktualisierung der Broker-Größe

Sie können Ihren MSK-Cluster bei Bedarf skalieren, indem Sie die Größe Ihrer Broker ändern, ohne Apache Kafka-Partitionen neu zuzuweisen. Wenn Sie die Größe Ihrer Broker ändern, haben Sie die Flexibilität, die Rechenkapazität Ihres MSK-Clusters an Änderungen Ihrer Workloads anzupassen, ohne Ihre Cluster-I/O zu unterbrechen. Amazon MSK verwendet dieselbe Broker-Größe für alle Broker in einem bestimmten Cluster.

In diesem Abschnitt wird beschrieben, wie Sie die Broker-Größe für Ihren MSK-Cluster aktualisieren. Sie können die Größe Ihres Cluster-Brokers von M5 oder T3 auf M7g oder von M7g auf M5 aktualisieren. Beachten Sie, dass die Migration zu einer kleineren Broker-Größe die Leistung und den maximal erreichbaren Durchsatz pro Broker verringern kann. Die Migration zu einem größeren Broker kann die Leistung steigern, kann aber auch mehr kosten.

Die Aktualisierung der Brokergröße erfolgt fortlaufend, während der Cluster läuft. Das bedeutet, dass Amazon MSK jeweils einen Broker herunterfährt, um das Broker-Size-Update durchzuführen. Informationen darüber, wie Sie einen Cluster während eines Broker-Size-Updates hochverfügbar machen können, finden Sie unter. the section called “Erstellen hochverfügbarer Cluster” Um mögliche Auswirkungen auf die Produktivität weiter zu reduzieren, können Sie das Broker-Size-Update in Zeiten mit geringem Datenverkehr durchführen.

Während eines Broker-Size-Updates können Sie weiterhin Daten produzieren und nutzen. Sie müssen jedoch warten, bis das Update abgeschlossen ist, bevor Sie Broker neu starten oder einen der unter Amazon-MSK-Vorgänge aufgeführten Aktualisierungsvorgänge aufrufen können.

Wenn Sie Ihren Cluster auf eine kleinere Broker-Größe aktualisieren möchten, empfehlen wir Ihnen, das Update zunächst auf einem Testcluster auszuprobieren, um zu sehen, wie es sich auf Ihr Szenario auswirkt.

> **Important**
>
> Sie können einen Cluster nicht auf eine kleinere Broker-Größe aktualisieren, wenn die Anzahl der Partitionen pro Broker die unter angegebene Höchstzahl überschreitetthe section called “ Die Größe Ihres Clusters anpassen: Anzahl der Partitionen pro Broker”.

Aktualisierung der Broker-Größe mit dem AWS Management Console

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
2. Wählen Sie den MSK-Cluster aus, für den Sie die Broker-Größe aktualisieren möchten.
3. Suchen Sie auf der Detailseite für den Cluster den Abschnitt Broker-Zusammenfassung und wählen Sie Brokergröße bearbeiten aus.
4. Wählen Sie die gewünschte Broker-Größe aus der Liste aus.
5. Speichern Sie die Änderungen.

Aktualisierung der Broker-Größe mit dem AWS CLI

1. Führen Sie den folgenden Befehl aus und *ClusterArn*ersetzen Sie ihn durch den Amazon-Ressourcennamen (ARN), den Sie bei der Erstellung Ihres Clusters erhalten haben. Wenn Ihnen der ARN für Ihren Cluster nicht vorliegt, finden Sie ihn, indem Sie alle Cluster auflisten. Weitere Informationen finden Sie unter the section called “Auflisten von Clustern”.

   Ersetzen Sie *Current-Cluster-Version* durch die aktuelle Version des Clusters und *TargetType*durch die neue Größe, die die Broker haben sollen. Weitere Informationen zu Broker-Größen finden Sie unter. the section called “Größen der Makler”

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

Nachfolgend finden Sie ein Beispiel für der Verwendung dieses Befehls.

```
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
```

Die Ausgabe dieses -Befehls sieht wie das folgende JSON-Beispiel aus.

```
{
    "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. Um das Ergebnis des `update-broker-type` Vorgangs zu erhalten, führen Sie den folgenden Befehl aus und ersetzen Sie *`ClusterOperationArn`* durch den ARN, den Sie in der Ausgabe des `update-broker-type` Befehls erhalten haben.

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

Die Ausgabe dieses `describe-cluster-operation`-Befehls sieht wie das folgende JSON-Beispiel aus.

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

Wenn `OperationState` den Wert „`UPDATE_IN_PROGRESS`“ aufweist, warten Sie eine Weile, bevor Sie den `describe-cluster-operation`-Befehl erneut ausführen.

Aktualisierung der Broker-Größe mithilfe der API

Informationen zum Aktualisieren der Broker-Größe mithilfe der API finden Sie unter UpdateBrokerTyp.

Sie können `UpdateBrokerType` die Größe Ihres Cluster-Brokers von M5 oder T3 auf M7g oder von M7g auf M5 aktualisieren.

Aktualisieren der Cluster-Konfiguration eines Amazon-MSK-Clusters

Um die Konfiguration eines Clusters aktualisieren zu können, sorgen Sie dafür, dass sich der Cluster im Status `ACTIVE` befindet. Sie müssen außerdem sicherstellen, dass die Anzahl der Partitionen pro Broker in Ihrem MSK-Cluster unter den in the section called “ Die Größe Ihres Clusters anpassen: Anzahl der Partitionen pro Broker” beschriebenen Grenzwerten liegt. Sie können die Konfiguration eines Clusters, der diese Grenzwerte überschreitet, nicht aktualisieren.

Informationen zur MSK-Konfiguration, einschließlich der Erstellung einer benutzerdefinierten Konfiguration, der Eigenschaften, die Sie aktualisieren können, und was passiert, wenn Sie die Konfiguration eines vorhandenen Clusters aktualisieren, finden Sie unter Konfiguration.

Aktualisierung der Konfiguration eines Clusters mithilfe des AWS CLI

1. Kopieren Sie das folgende JSON und speichern Sie es in einer Datei. Benennen Sie die Datei `configuration-info.json`. *`ConfigurationArn`*Ersetzen Sie durch den Amazon-Ressourcennamen (ARN) der Konfiguration, die Sie für die Aktualisierung des Clusters verwenden möchten. Die ARN-Zeichenfolge muss in Anführungszeichen im folgenden JSON erfolgen.

Ersetzen Sie *Configuration-Revision* durch die Revision der zu verwendenden Konfiguration. Konfigurationsrevisionen sind Ganzzahlen, die bei 1 beginnen. Diese Ganzzahl darf im folgenden JSON nicht von Anführungszeichen umgeben sein.

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

2. Führen Sie den folgenden Befehl aus und *ClusterArn*ersetzen Sie ihn durch den ARN, den Sie bei der Erstellung Ihres Clusters erhalten haben. Wenn Ihnen der ARN für Ihren Cluster nicht vorliegt, finden Sie ihn, indem Sie alle Cluster auflisten. Weitere Informationen finden Sie unter the section called “Auflisten von Clustern”.

   Ersetzen Sie *Pfad-zu-config-info-Datei* durch den Pfad zu Ihrer Konfigurationsinfodatei. Wenn Sie die im vorherigen Schritt erstellte `configuration-info.json`-Datei benannt und sie im aktuellen Verzeichnis gespeichert haben, lautet der *Pfad-zu-Config-Info-Datei* `configuration-info.json`.

   Ersetzen Sie *Aktuelle-Cluster-Version* durch die aktuelle Version des Clusters.

   > **Important**
   >
   > Cluster-Versionen sind keine einfachen Ganzzahlen. Verwenden Sie den Befehl DescribeClusteroperation oder describe-cluster, um die aktuelle Version des Clusters AWS CLI zu ermitteln. `KTVPDKIKX0DER` ist ein Beispiel für eine Version.

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

   Nachfolgend finden Sie ein Beispiel für der Verwendung dieses Befehls.

```
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"
```

Die Ausgabe dieses `update-cluster-configuration`-Befehls sieht wie das folgende JSON-Beispiel aus.

```
{
    "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. Um das Ergebnis des `update-cluster-configuration` Vorgangs zu erhalten, führen Sie den folgenden Befehl aus und ersetzen Sie *`ClusterOperationArn`* durch den ARN, den Sie in der Ausgabe des `update-cluster-configuration` Befehls erhalten haben.

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

Die Ausgabe dieses `describe-cluster-operation`-Befehls sieht wie das folgende JSON-Beispiel aus.

```
{
    "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 dieser Ausgabe hat `OperationType` den Wert „`UPDATE_CLUSTER_CONFIGURATION`“. Wenn `OperationState` den Wert „`UPDATE_IN_PROGRESS`“ aufweist, warten Sie eine Weile, bevor Sie den `describe-cluster-operation`-Befehl erneut ausführen.

Aktualisieren der Konfiguration eines Clusters mithilfe der API

Informationen zur Verwendung der API zum Aktualisieren der Konfiguration eines Clusters finden Sie unter UpdateClusterKonfiguration.

Erweitern eines Amazon-MSK-Clusters

Verwenden Sie diesen Amazon-MSK-Vorgang, wenn Sie die Anzahl der Broker in Ihrem MSK-Cluster erhöhen möchten. Um einen Cluster zu erweitern, stellen Sie sicher, dass er sich im Status `ACTIVE` befindet.

> **Important**
>
> Wenn Sie einen MSK Cluster erweitern möchten, stellen Sie sicher, dass Sie diesen Amazon-MSK-Vorgang verwenden. Versuchen Sie nicht, Broker ohne Verwendung dieses Vorgangs einem Cluster hinzuzufügen.

Informationen zum Neuausgleich von Partitionen nach dem Hinzufügen von Brokern zu einem Cluster finden Sie unter the section called “Neuzuweisung von Partitionen”.

Erweiterung eines Clusters mit dem AWS Management Console

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
2. Wählen Sie den MSK-Cluster aus, dessen Broker-Anzahl erhöht werden soll.
3. Wählen Sie auf der Seite der Cluster-Details die Schaltfläche Bearbeiten neben der Überschrift Broker-Details auf Cluster-Ebene.
4. Geben Sie die Anzahl der Broker ein, die dem Cluster pro Availability Zone zur Verfügung stehen sollen, und wählen Sie dann Änderungen speichern.

Erweiterung eines Clusters mit dem AWS CLI

1. Führen Sie den folgenden Befehl aus und *ClusterArn*ersetzen Sie ihn durch den Amazon-Ressourcennamen (ARN), den Sie bei der Erstellung Ihres Clusters erhalten haben. Wenn Ihnen der ARN für Ihren Cluster nicht vorliegt, finden Sie ihn, indem Sie alle Cluster auflisten. Weitere Informationen finden Sie unter the section called “Auflisten von Clustern”.

   Ersetzen Sie *Aktuelle-Cluster-Version* durch die aktuelle Version des Clusters.

   > **Important**
   >
   > Cluster-Versionen sind keine einfachen Ganzzahlen. Um die aktuelle Version des Clusters zu finden, verwenden Sie den DescribeClusterBefehl operation oder describe-cluster AWS CLI . `KTVPDKIKX0DER` ist ein Beispiel für eine Version.

   Der Parameter *Target-Number-of-Brokers* stellt die Gesamtzahl der Broker-Knoten für den Cluster dar, wenn dieser Vorgang erfolgreich abgeschlossen wird. Der Wert, den Sie für *Target-Number-of-Brokers* angeben, muss eine Ganzzahl sein, die größer ist als die aktuelle Anzahl der Broker im Cluster. Sie muss auch ein Vielfaches der Anzahl der Availability Zones sein.

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

   Die Ausgabe dieses `update-broker-count`-Vorgangs sieht wie das folgende JSON aus.

   ```
   {

       "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. Um das Ergebnis des `update-broker-count` Vorgangs zu erhalten, führen Sie den folgenden Befehl aus und ersetzen Sie *ClusterOperationArn* durch den ARN, den Sie in der Ausgabe des `update-broker-count` Befehls erhalten haben.

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

Die Ausgabe dieses `describe-cluster-operation`-Befehls sieht wie das folgende JSON-Beispiel aus.

```
{
    "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 dieser Ausgabe hat `OperationType` den Wert „`INCREASE_BROKER_COUNT`“. Wenn `OperationState` den Wert „`UPDATE_IN_PROGRESS`“ aufweist, warten Sie eine Weile, bevor Sie den `describe-cluster-operation`-Befehl erneut ausführen.

Erweitern eines Clusters mithilfe der API

Informationen zur Erhöhung der Anzahl der Broker in einem Cluster, die die API verwenden, finden Sie unter UpdateBrokerAnzahl.

Einen Broker aus einem Amazon MSK-Cluster entfernen

Verwenden Sie diesen Amazon MSK-Vorgang, wenn Sie Broker aus den von Amazon Managed Streaming for Apache Kafka (MSK) bereitgestellten Clustern entfernen möchten. Sie können die

Speicher- und Rechenkapazität Ihres Clusters reduzieren, indem Sie Gruppen von Brokern entfernen, ohne dass dies Auswirkungen auf die Verfügbarkeit, das Risiko der Datenbeständigkeit oder eine Unterbrechung Ihrer Datenstreaming-Anwendungen hat.

Sie können Ihrem Cluster weitere Broker hinzufügen, um den Anstieg des Datenverkehrs zu bewältigen, und Broker entfernen, wenn der Verkehr nachlässt. Mit den Funktionen zum Hinzufügen und Entfernen von Brokern können Sie Ihre Clusterkapazität optimal nutzen und Ihre MSK-Infrastrukturkosten optimieren. Durch das Entfernen von Brokern haben Sie die Kontrolle über die vorhandene Clusterkapazität auf Broker-Ebene, um sie an Ihre Workload-Anforderungen anzupassen und eine Migration zu einem anderen Cluster zu vermeiden.

Verwenden Sie die AWS Konsole, die Befehlszeilenschnittstelle (CLI), das SDK oder, AWS CloudFormation um die Anzahl der Broker Ihres bereitgestellten Clusters zu reduzieren. MSK wählt die Broker aus, auf denen sich keine Partitionen befinden (außer bei kanarischen Themen), und verhindert, dass Anwendungen Daten an diese Broker senden. Gleichzeitig werden diese Broker sicher aus dem Cluster entfernt.

Sie sollten einen Broker pro Availability Zone entfernen, wenn Sie den Speicher- und Rechenaufwand eines Clusters reduzieren möchten. Sie können beispielsweise zwei Broker aus einem Cluster mit zwei Availability Zones oder drei Broker aus einem Cluster mit drei Availability Zones in einem einzigen Broker-Entfernungsvorgang entfernen.

Informationen dazu, wie Sie Partitionen neu verteilen können, nachdem Sie Broker aus einem Cluster entfernt haben, finden Sie unterthe section called “Neuzuweisung von Partitionen”.

Sie können Broker aus allen M5- und M7g-basierten, von MSK bereitgestellten Clustern entfernen, unabhängig von der Instanzgröße.

Das Entfernen von Brokern wird in den Kafka-Versionen 2.8.1 und höher unterstützt, auch in Clustern im KraFT-Modus.

Themen

- Bereiten Sie sich darauf vor, Broker zu entfernen, indem Sie alle Partitionen entfernen
- Entfernen Sie einen Broker mit der AWS Management Console
- Entfernen Sie einen Broker mit der AWS CLI
- Entfernen Sie einen Broker mit der API AWS

Bereiten Sie sich darauf vor, Broker zu entfernen, indem Sie alle Partitionen entfernen

Bevor Sie mit dem Broker-Entfernungsprozess beginnen, verschieben Sie zunächst alle Partitionen mit Ausnahme der Partitionen für Themen `__amazon_msk_canary` und `__amazon_msk_canary_state` für die Broker, die Sie entfernen möchten. Dies sind interne Themen, die Amazon MSK für Cluster-Integritäts- und Diagnosemetriken erstellt.

Sie können Kafka-Admin-APIs oder Cruise Control verwenden, um Partitionen auf andere Broker zu verschieben, die Sie im Cluster behalten möchten. Siehe Partitionen neu zuweisen.

Beispielprozess zum Entfernen von Partitionen

Dieser Abschnitt ist ein Beispiel dafür, wie Sie Partitionen aus dem Broker entfernen können, den Sie entfernen möchten. Angenommen, Sie haben einen Cluster mit 6 Brokern, 2 Brokern in jeder AZ, und er hat vier Themen:

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

1. Erstellen Sie einen Client-Computer, wie unter Client-Computer erstellen beschrieben.
2. Führen Sie nach der Konfiguration des Client-Computers den folgenden Befehl aus, um alle verfügbaren Themen in Ihrem Cluster aufzulisten.

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

   In diesem Beispiel sehen wir vier Themennamen: `__amazon_msk_canary__consumer_offsets`,`__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-c657f7e4ff32-2`, und`msk-brk-rmv`.
3. Erstellen Sie eine JSON-Datei, die `topics.json` auf dem Client-Computer aufgerufen wird, und fügen Sie alle Benutzerthemennamen wie im folgenden Codebeispiel hinzu. Sie müssen den `__amazon_msk_canary` Themennamen nicht angeben, da es sich um ein vom Service verwaltetes Thema handelt, das bei Bedarf automatisch verschoben wird.

```
{
"topics": [
{"topic": "msk-brk-rmv"},
{"topic": "__consumer_offsets"},
{"topic": "__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-
c657f7e4ff32-2"}
],
"version":1
}
```

4. Führen Sie den folgenden Befehl aus, um einen Vorschlag zum Verschieben von Partitionen auf nur 3 von 6 Brokern im Cluster zu generieren.

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

5. Erstellen Sie eine Datei mit dem Namen `reassignment-file.json` und kopieren `proposed partition reassignment configuration` Sie den Befehl, den Sie vom obigen Befehl erhalten haben.
6. Führen Sie den folgenden Befehl aus, um Partitionen zu verschieben, die Sie in der angegeben haben`reassignment-file.json`.

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

Die Ausgabe sieht folgendermaßen oder ähnlich aus:

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

7. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob alle Partitionen verschoben wurden.

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

Die Ausgabe sieht folgendermaßen oder ähnlich aus. Überwachen Sie den Status, bis alle Partitionen in den von Ihnen angeforderten Themen erfolgreich neu zugewiesen wurden:

```
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. Wenn der Status anzeigt, dass die Neuzuweisung der Partitionen für jede Partition abgeschlossen ist, überwachen Sie die `UserPartitionExists` Metriken fünf Minuten lang, um sicherzustellen, dass sie `0` für die Broker angezeigt werden, von denen Sie die Partitionen verschoben haben. Nachdem Sie dies bestätigt haben, können Sie damit fortfahren, den Broker aus dem Cluster zu entfernen.

Entfernen Sie einen Broker mit der AWS Management Console

Um Broker mit der AWS Management Console zu entfernen

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msks/.
2. Wählen Sie den MSK-Cluster aus, der Broker enthält, die Sie entfernen möchten.
3. Klicken Sie auf der Seite mit den Cluster-Details auf die Schaltfläche Aktionen und wählen Sie die Option Anzahl der Broker bearbeiten aus.
4. Geben Sie die Anzahl der Broker ein, die der Cluster pro Availability Zone haben soll. In der Konsole wird die Anzahl der Broker in den Availability Zones zusammengefasst, die entfernt werden. Stellen Sie sicher, dass dies das ist, was Sie wollen.
5. Wählen Sie Änderungen speichern aus.

Um ein versehentliches Entfernen von Brokern zu verhindern, werden Sie in der Konsole aufgefordert, zu bestätigen, dass Sie Broker löschen möchten.

Entfernen Sie einen Broker mit der AWS CLI

Führen Sie den folgenden Befehl aus und `ClusterArn` ersetzen Sie ihn durch den Amazon-Ressourcennamen (ARN), den Sie bei der Erstellung Ihres Clusters erhalten haben. Wenn Ihnen der ARN für Ihren Cluster nicht vorliegt, finden Sie ihn, indem Sie alle Cluster auflisten. Weitere Informationen finden Sie unter Amazon MSK-Cluster auflisten. `Current-Cluster-Version`Durch die aktuelle Version des Clusters ersetzen.

> **Important**
>
> Cluster-Versionen sind keine einfachen Ganzzahlen. Verwenden Sie den Befehl DescribeClusteroperation oder describe-cluster, um die aktuelle Version des Clusters AWS CLI zu finden. KTVPDKIKX0DER ist ein Beispiel für eine Version.

Der Parameter *Target-Number-of-Brokers* stellt die Gesamtzahl der Broker-Knoten für den Cluster dar, wenn dieser Vorgang erfolgreich abgeschlossen wird. Der Wert, den Sie für *Target-Number-of-Brokers* angeben, muss eine ganze Zahl sein, die kleiner ist als die aktuelle Anzahl von Brokern im Cluster. Sie muss auch ein Vielfaches der Anzahl der Availability Zones sein.

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

Die Ausgabe dieses `update-broker-count`-Vorgangs sieht wie das folgende JSON aus.

```
{
"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 dieser Ausgabe hat `OperationType` den Wert „`DECREASE_BROKER_COUNT`“. Wenn `OperationState` den Wert „`UPDATE_IN_PROGRESS`“ aufweist, warten Sie eine Weile, bevor Sie den `describe-cluster-operation`-Befehl erneut ausführen.

Entfernen Sie einen Broker mit der API AWS

Informationen zum Entfernen von Brokern in einem Cluster mithilfe der API finden Sie unter UpdateBrokerAnzahl in der Amazon Managed Streaming for Apache Kafka API-Referenz.

Aktualisieren der Sicherheitseinstellungen eines Clusters

Verwenden Sie diesen Amazon-MSK-Vorgang, um die Authentifizierungs- und Client-Broker-Verschlüsselungseinstellungen Ihres MSK-Clusters zu aktualisieren. Sie können auch die Private Security Authority aktualisieren, die zum Signieren von Zertifikaten für die gegenseitige TLS-Authentifizierung verwendet wird. Sie können die Verschlüsselungseinstellung im Cluster (Broker-to-Broker) nicht ändern.

Der Cluster muss sich in dem Status `ACTIVE` befinden, damit Sie seine Sicherheitseinstellungen aktualisieren können.

Wenn Sie die Authentifizierung mit IAM, SASL oder TLS aktivieren, müssen Sie auch die Verschlüsselung zwischen Clients und Brokern aktivieren. Die folgende Tabelle zeigt die möglichen Kombinationen.

Authentifizierung	Verschlüsselungsoptionen für Client-Broker	Broker-Broker-Verschlüsselung
Nicht authentifiziert	TLS, PLAINTEXT, TLS_PLAINTEXT	Kann Ein oder Aus sein.
mTLS	TLS, TLS_PLAINTEXT	Muss Ein sein.
SASL/SCRAM	TLS	Muss Ein sein.
SASL/IAM	TLS	Muss Ein sein.

Wenn die Client-Broker-Verschlüsselung auf `TLS_PLAINTEXT` und die Client-Authentifizierung auf `mTLS` eingestellt sind, erstellt Amazon MSK zwei Arten von Listenern, mit denen sich Clients verbinden können: einen Listener, mit dem sich Clients mithilfe von mTLS-Authentifizierung mit TLS-Verschlüsselung verbinden können, und einen anderen, für Clients, die sich ohne Authentifizierung oder Verschlüsselung (Klartext) verbinden können.

Weitere Informationen zu den Sicherheitseinstellungen finden Sie unter Sicherheit.

Aktualisierung der Sicherheitseinstellungen eines Clusters mithilfe der AWS Management Console

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
2. Wählen Sie den MSK-Cluster, den Sie aktualisieren möchten.
3. Wählen Sie im Abschnitt Sicherheitseinstellungen die Option Bearbeiten.
4. Wählen Sie die gewünschten Authentifizierungs- und Verschlüsselungseinstellungen für den Cluster aus danach Änderungen speichern.

Aktualisierung der Sicherheitseinstellungen eines Clusters mithilfe der AWS CLI

1. Erstellen Sie eine JSON-Datei, die die Verschlüsselungseinstellungen enthält, die der Cluster haben soll. Im Folgenden wird ein Beispiel gezeigt.

   > **Note**
   >
   > Sie können nur die Client-Broker-Verschlüsselungseinstellung aktualisieren. Sie können die Verschlüsselungseinstellung im Cluster (broker-to-broker) nicht ändern.

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

2. Erstellen Sie eine JSON-Datei, die die Authentifizierungseinstellungen enthält, die der Cluster haben soll. Im Folgenden wird ein Beispiel gezeigt.

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

3. Führen Sie den folgenden AWS CLI Befehl aus:

   ```
   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
   ```

   Die Ausgabe dieses `update-security`-Vorgangs sieht wie das folgende JSON aus.

```
{

    "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. Um den Status des `update-security` Vorgangs zu überprüfen, führen Sie den folgenden Befehl aus und ersetzen Sie *`ClusterOperationArn`* durch den ARN, den Sie in der Ausgabe des `update-security` Befehls erhalten haben.

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

Die Ausgabe dieses `describe-cluster-operation`-Befehls sieht wie das folgende JSON-Beispiel aus.

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

Wenn `OperationState` den Wert `PENDING` oder `UPDATE_IN_PROGRESS` aufweist, warten Sie eine Weile, bevor Sie den Befehl `describe-cluster-operation` erneut ausführen.

Aktualisieren der Sicherheitseinstellungen eines Clusters mithilfe der API

Informationen zum Aktualisieren der Sicherheitseinstellungen für einen Cluster mithilfe der API finden Sie unter UpdateSecurity.

> **Note**
>
> Die AWS CLI und API-Operationen zum Aktualisieren der Sicherheitseinstellungen eines Clusters sind idempotent. Das heißt, wenn Sie das Sicherheitsupdate aufrufen und eine Authentifizierungs- oder Verschlüsselungseinstellung angeben, die der aktuellen Einstellung des Clusters entspricht, ändert sich diese Einstellung nicht.

Neustarten eines Brokers für einen Amazon-MSK-Cluster

Verwenden Sie diesen Amazon-MSK-Vorgang, wenn Sie einen Broker in Ihrem MSK-Cluster neustarten möchten. Um einen Broker für einen Cluster neu zu starten, stellen Sie sicher, dass sich der Cluster im `ACTIVE` Status befindet.

Der Amazon-MSK-Service kann die Broker für Ihren MSK-Cluster während der Systemwartung neu starten, z. B. beim Patchen oder bei Versions-Upgrades. Wenn Sie einen Broker manuell neu starten, können Sie die Ausfallssicherheit Ihrer Kafka-Clients testen, um festzustellen, wie sie auf die Systemwartung reagieren.

Neustarten eines Brokers mit dem AWS Management Console

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
2. Wählen Sie den MSK-Cluster aus, dessen Broker neu gestartet werden soll.
3. Scrollen Sie nach unten zum Abschnitt Broker-Details und wählen Sie den Broker aus, den Sie neu starten möchten.
4. Wählen Sie die Schaltfläche Broker neu starten.

Neustart eines Brokers mit dem AWS CLI

1. Führen Sie den folgenden Befehl aus und *`ClusterArn`*ersetzen Sie ihn durch den Amazon-Ressourcennamen (ARN), den Sie bei der *`BrokerId`*Erstellung Ihres Clusters erhalten haben, und durch die ID des Brokers, den Sie neu starten möchten.

> **Note**
>
> Der `reboot-broker`-Vorgang unterstützt jeweils nur den Neustart eines Brokers.

Wenn Ihnen der ARN für Ihren Cluster nicht vorliegt, finden Sie ihn, indem Sie alle Cluster auflisten. Weitere Informationen finden Sie unter the section called “Auflisten von Clustern”.

Wenn Sie die Broker-IDs für Ihren Cluster nicht haben, können Sie sie finden, indem Sie die Broker-Knoten auflisten. Weitere Informationen finden Sie unter list-nodes.

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

Die Ausgabe dieses `reboot-broker`-Vorgangs sieht wie das folgende JSON aus.

```
{

    "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. Um das Ergebnis des `reboot-broker` Vorgangs zu erhalten, führen Sie den folgenden Befehl aus und *`ClusterOperationArn`*ersetzen Sie ihn durch den ARN, den Sie in der Ausgabe des `reboot-broker` Befehls erhalten haben.

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

Die Ausgabe dieses `describe-cluster-operation`-Befehls sieht wie das folgende JSON-Beispiel aus.

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

Wenn der Neustart-Vorgang abgeschlossen ist, ist `OperationState` `REBOOT_COMPLETE`.

Neustarten eines Brokers mit der API

Informationen zum Neustarten eines Brokers in einem Cluster mithilfe der API finden Sie unter RebootBroker.

Auswirkungen von Broker-Neustarts während Patches und anderen Wartungsarbeiten

In regelmäßigen Abständen aktualisiert Amazon MSK die Software auf Ihren Brokern. Diese Updates haben keine Auswirkungen auf die Schreib- und Lesevorgänge Ihrer Anwendungen, sofern Sie die bewährten Methoden befolgen.

Amazon MSK verwendet fortlaufende Updates für Software, um die hohe Verfügbarkeit Ihrer Cluster aufrechtzuerhalten. Während dieses Vorgangs werden die Broker nacheinander neu gestartet, und Kafka überträgt die Leitung automatisch auf einen anderen Online-Broker. Kafka-Clients verfügen über integrierte Mechanismen, die den Wechsel in der Führung der Partitionen automatisch erkennen und weiterhin Daten in einen MSK-Cluster schreiben und lesen.

Wenn ein Broker offline geht, ist es normal, dass auf Ihren Clients vorübergehende Verbindungsfehler auftreten. Außerdem werden Sie für einen kurzen Zeitraum (bis zu 2 Minuten, in der Regel weniger) einige Spitzen der p99-Lese- und Schreiblatenz beobachten (typischerweise hohe Millisekunden, bis zu ~2 Sekunden). Diese Spitzenwerte sind zu erwarten und werden dadurch verursacht, dass der Kunde erneut eine Verbindung zu einem neuen führenden Broker herstellt. Sie wirken sich nicht auf Ihre Produktion oder Ihren Verbrauch aus und werden nach der erneuten Verbindung wieder behoben.

Sie werden auch einen Anstieg der Metrik feststellen`UnderReplicatedPartitions`, was zu erwarten ist, da die Partitionen auf dem heruntergefahrenen Broker keine Daten mehr replizieren. Dies hat keine Auswirkungen auf die Schreib- und Lesevorgänge der Anwendungen, da Replikate für diese Partitionen, die auf anderen Brokern gehostet werden, die Anfragen nun bearbeiten.

Wenn der Broker nach dem Softwareupdate wieder online ist, muss er die Nachrichten „catch“, die während des Offline-Betriebs generiert wurden. Während der Nachholphase können Sie auch einen Anstieg der Auslastung des Volumendurchsatzes und der CPU beobachten. Diese sollten keine Auswirkungen auf Schreib- und Lesevorgänge in den Cluster haben, wenn Sie über genügend CPU-, Arbeitsspeicher-, Netzwerk- und Volume-Ressourcen auf Ihren Brokern verfügen.

Markieren eines Amazon-MSK-Clusters

Sie können einer Amazon-MSK-Ressource, z. B. einem MSK-Cluster, Ihre eigenen Metadaten in Form von Tags zuweisen. Ein Tag ist ein Schlüssel-Wert-Paar, das Sie für die Ressource definieren. Die Verwendung von Tags ist eine einfache und dennoch leistungsstarke Möglichkeit, AWS Ressourcen zu verwalten und Daten, einschließlich Rechnungsdaten, zu organisieren.

Themen

- Grundlagen zu Tags (Markierungen)
- Verfolgen der Kosten mithilfe von Markierungen
- Tag-Einschränkungen
- Markieren von Ressourcen mithilfe der Amazon-MSK-API

Grundlagen zu Tags (Markierungen)

Sie können die Amazon-MSK-API verwenden, um die folgenden Aufgaben auszuführen:

- Einer Amazon-MSK-Ressource Tags hinzufügen.
- Die Tags für eine Amazon-MSK-Ressource auflisten.
- Tags von einer Amazon-MSK-Ressource entfernen.

Sie können mit Tags Ihre Amazon-MSK-Ressourcen kategorisieren. Sie können Ihre Amazon-MSK-Cluster beispielsweise nach Zweck, Besitzer oder Umgebung kategorisieren. Da Sie für jeden Tag den Schlüssel und Wert definieren, können Sie eine auf benutzerdefinierte Reihe von Kategorien

anlegen, die Ihren jeweiligen Anforderungen gerecht wird. Sie könnten zum Beispiel eine Reihe von Tags definieren, mit der Sie Cluster nach Besitzer und zugehöriger Anwendung nachverfolgen können.

Im Folgenden sehen Sie verschiedene Beispiele für Tags:

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

Verfolgen der Kosten mithilfe von Markierungen

Sie können Tags verwenden, um Ihre AWS Kosten zu kategorisieren und nachzuverfolgen. Wenn Sie Tags auf Ihre AWS Ressourcen anwenden, einschließlich Amazon MSK-Clustern, enthält Ihr AWS Kostenzuordnungsbericht die Nutzung und die Kosten, die nach Tags zusammengefasst sind. Sie können die Kosten für mehrere Services organisieren, indem Sie Tags anwenden, die geschäftliche Kategorien (wie Kostenstellen, Anwendungsnamen oder Eigentümer) darstellen. Weitere Informationen finden Sie unter Verwenden von Kostenzuordnungs-Tags für benutzerdefinierte Fakturierungsberichte im AWS Billing -Benutzerhandbuch.

Tag-Einschränkungen

Für Tags in Amazon MSK gelten die folgenden Einschränkungen.

Grundlegende Einschränkungen

- Die maximale Anzahl an Tags pro Ressource beträgt 50.
- Bei Tag-Schlüsseln und -Werten wird zwischen Groß- und Kleinschreibung unterschieden.
- Sie können Tags für eine gelöschte Ressource nicht ändern oder bearbeiten.

Einschränkungen für Tag-Schlüssel

- Jeder Tag-Schlüssel muss einmalig sein. Wenn Sie einen Tag mit einem Schlüssel hinzufügen, der bereits verwendet wird, wird das vorhandene Schlüssel-Wert-Paar durch den neuen Tag überschrieben.

- Sie können einen Tag-Schlüssel nicht mit `aws:` beginnen, da dieses Präfix für die Verwendung durch AWS reserviert ist. AWS erstellt in Ihrem Namen Tags, die mit diesem Präfix beginnen, Sie können diese jedoch nicht bearbeiten oder löschen.
- Tag-Schlüssel müssen zwischen 1 und 128 Unicode-Zeichen lang sein.
- Tag-Schlüssel müssen die folgenden Zeichen enthalten: Unicode-Zeichen, Ziffern, Leerzeichen sowie die folgenden Sonderzeichen: `_ . / = + - @`.

Einschränkungen für den Tag-Wert

- Tag-Werte müssen zwischen 0 und 255 Unicode-Zeichen lang sein.
- Tag-Werte können leer sein. Ansonsten müssen sie die folgenden Zeichen enthalten: Unicode-Zeichen, Ziffern, Leerzeichen und eines der folgenden Sonderzeichen: `_ . / = + - @`.

Markieren von Ressourcen mithilfe der Amazon-MSK-API

Mit den folgenden Vorgängen können Sie eine Amazon-MSK-Ressource mit einem Tag kennzeichnen bzw. eine Kennzeichnung aufheben oder den aktuellen Satz von Tags für eine Ressource auflisten:

- ListTagsForResource
- TagResource
- UntagResource

Amazon-MSK-Konfiguration

Amazon Managed Streaming for Apache Kafka bietet eine Standardkonfiguration für Broker, Themen und ZooKeeper Apache-Knoten. Ebenso können Sie benutzerdefinierte Konfigurationen erstellen und sie verwenden, um neue MSK-Cluster zu erstellen oder vorhandene Cluster zu aktualisieren. Eine MSK-Konfiguration besteht aus einer Reihe von Eigenschaften und den entsprechenden Werten.

Themen

Benutzerdefinierte MSK-Konfigurationen

Mit Amazon MSK können Sie eine benutzerdefinierte MSK-Konfiguration erstellen, in der Sie die folgenden Eigenschaften festlegen. Eigenschaften, die Sie nicht explizit festlegen, erhalten die in the section called “Standardkonfiguration” festgelegten Werte. Weitere Informationen zu Konfigurationseigenschaften finden Sie unter Apache Kafka Configuration.

Apache-Kafka-Konfigurationseigenschaften

Name	Beschreibung
allow.everyone.if.no.acl.found	Wenn Sie diese Eigenschaft auf `false` setzen möchten, stellen Sie zunächst sicher, dass Sie Apache-Kafka-ACLs für Ihren Cluster definiere n. Wenn Sie diese Eigenschaft auf `false` setzen und Sie nicht zuerst Apache-Kafka-ACLs definieren, verlieren Sie den Zugriff auf den Cluster. In diesem Fall können Sie die Konfiguration erneut aktualisieren und diese Eigenschaft auf `true` setzen, um wieder Zugriff auf den Cluster zu erhalten.

Name	Beschreibung
auto.create.topics.enable	Aktiviert die automatische Erstellung von Themen auf dem Server.
compression.type	Der endgültige Komprimierungstyp für ein bestimmtes Thema. Sie können diese Eigenschaft auf die Standard-Komprimierungscodecs (`gzip`, `snappy`, `lz4` und `zstd`) festlegen. Akzeptiert zusätzlich `uncompressed`. Dieser Wert entspricht keiner Komprimierung. Wenn Sie den Wert auf `producer` setzen, bedeutet dies, dass der ursprüngliche Komprimierungs-Codec beibehalten wird, den der Produzent festlegt.
connections.max.idle.ms	Timeout bei inaktiven Verbindungen in Millisekunden. Die Threads des Server-Socket-Prozessors schließen die Verbindungen, die länger als den von Ihnen für diese Eigenschaft festgelegten Wert inaktiv sind.
default.replication.factor	Der Standardreplikationsfaktor für automatisch erstellte Themen.
delete.topic.enable	Aktiviert den Vorgang zum Löschen von Themen. Wenn Sie diese Einstellung deaktivieren, können Sie ein Thema nicht über das Admin-Tool löschen.
group.initial.rebalance.delay.ms	Die Zeit, die der Gruppenkoordinator darauf wartet, dass mehr Verbraucher einer neuen Gruppe beitreten, bevor der erste Neuausgleich durchgeführt wird. Eine längere Verzögerung bedeutet potenziell wenigere Neuausgleiche, erhöht aber die Zeit bis zum Beginn der Verarbeitung.

Name	Beschreibung
group.max.session.timeout.ms	Maximales Sitzungs-Timeout für registrierte Konsumenten. Längere Timeouts verschaff en Verbrauchern mehr Zeit für die Verarbeit ung von Nachrichten zwischen Heartbeats, sie führen aber auch zu einer längeren Fehlererk ennungszeit.
group.min.session.timeout.ms	Minimale Sitzungs-Timeout für registrierte Konsumenten. Kürzere Timeouts führen zu einer schnelleren Fehlererkennung und häufigeren Verbraucher-Heartbeats, was Broker-Ressourcen überfordern kann. Dies kann die Broker-Ressourcen überfordern.
leader.imbalance.per.broker.percentage	Das Verhältnis des zulässigen Führungsu ngleichgewichts pro Broker. Der Controller löst einen Führungsausgleich aus, wenn er diesen Wert pro Broker übersteigt. Dieser Wert wird in Prozent angegeben.
log.cleaner.delete.retention.ms	Zeitraum, in dem Apache Kafka gelöschte Datensätze beibehalten soll. Der Mindestwert ist 0.

Name	Beschreibung
log.cleaner.min.cleanable.ratio	Diese Konfigurationseigenschaft kann Werte zwischen 0 und 1 haben. Dieser Wert bestimmt, wie oft der Protokollkomprimierer versucht, das Protokoll zu bereinigen (wenn die Protokollkomprimierung aktiviert ist). Standardmäßig vermeidet Apache Kafka die Bereinigung eines Protokolls, wenn mehr als 50 % des Protokolls komprimiert wurden. Dieses Verhältnis begrenzt den maximalen Speicherplatz, den das Protokoll mit Duplikaten verschwendet (bei 50 % bedeutet dies, dass höchstens 50 % des Protokolls Duplikate sein könnten). Bei einem größeren Verhältnis sind Bereinigungen häufiger und effizienter, aber es wird auch mehr Speicherplatz im Protokoll benötigt.
log.cleanup.policy	Die Standard-Bereinigungsrichtlinie für Segmente außerhalb des Aufbewahrungsfensters. Eine durch Kommata getrennte Liste gültiger Richtlinien. Gültige Richtlinien sind `delete` und `compact`. Für Cluster mit aktiviertem gestaffelter Speicherung gilt nur die Richtlinie `delete`.
log.flush.interval.messages	Anzahl der Nachrichten, die auf einer Protokollpartition gesammelt werden, bevor Nachrichten auf den Datenträger geschrieben werden.

Name	Beschreibung
log.flush.interval.ms	Maximale Zeit in Millisekunden, in der eine Nachricht in einem beliebigen Thema im Speicher aufbewahrt wird, bevor sie auf die Festplatte geschrieben wird. Wenn Sie diesen Wert nicht festlegen, wird der Wert in log.flush.scheduler.interval.ms verwendet. Der Mindestwert ist 0.
log.message.timestamp.difference.max.ms	Der maximale Zeitunterschied zwischen dem Zeitstempel beim Empfang einer Nachricht durch den Broker und dem in der Nachricht angegebenen Zeitstempel. Bei log.messa ge.timestamp.type= wird eine Nachricht zurückgewiesenCreateTime, wenn der Zeitstempelunterschied diesen Schwellen wert überschreitet. Diese Konfiguration wird LogAppend ignoriert, wenn log.message.timest amp.type= Zeit.
log.message.timestamp.type	Gibt an, wenn der Zeitstempel in der Nachricht die Erstellungszeit der Nachricht oder die Anfügezeit des Protokolls widerspiegelt. Die zulässigen Werte sind `CreateTime` und `LogAppendTime` .
log.retention.bytes	Maximale Größe des Protokolls vor dem Löschen.
log.retention.hours	Anzahl der Stunden, die eine Protokolldatei vor dem Löschen aufbewahrt werden muss, tertiär zur Eigenschaft log.retention.ms.

Name	Beschreibung
log.retention.minutes	Anzahl der Minuten, in denen eine Protokoll datei vor dem Löschen aufbewahrt wird, sekundär zur Eigenschaft log.retention.ms. Wenn Sie diesen Wert nicht festlegen, wird der Wert in log.retention.hours verwendet.
log.retention.ms	Anzahl der Millisekunden, die eine Protokoll datei vor dem Löschen aufbewahrt wird (in Millisekunden). Wenn der Wert nicht festgeleg t ist, wird der Wert in log.retention.minutes verwendet.
log.roll.ms	Maximale Zeit, bis ein neues Protokollsegment bereitgestellt wird (in Millisekunden). Wenn Sie diesen Wert nicht festlegen, wird der Wert in log.roll.hours verwendet. Der Mindestwert für diese Eigenschaft ist 1.
log.segment.bytes	Maximale Größe einer einzelnen Protokolldatei.
max.incremental.fetch.session.cache.slots	Maximale Anzahl inkrementeller Abrufsitz ungen, die beibehalten werden.

Name	Beschreibung
message.max.bytes	Die größte von Kafka unterstützte Protokoll-Batch-Größe. Wenn Sie diesen Wert erhöhen und Verbraucher älter als 0.10.2 vorhanden sind, müssen Sie auch die Abrufgröße der Verbraucher erhöhen, damit sie diese großen Datensatz-Batch abrufen können. In der neuesten Nachrichtenformat-Version werden Datensätze aus gründen der Effizienz immer in Batches gruppiert. In früheren Nachrichtenformat-Versionen werden nicht komprimierte Datensätze nicht in Batches gruppiert und diese Beschränkung gilt in diesem Fall nur für einen einzelnen Datensatz. Sie können dies pro Thema mit der Konfigura tion auf Themenebene max.message.bytes festlegen.

Name	Beschreibung
min.insync.replicas	Wenn ein Produzent acks auf `"all"` (oder `"-1"`) setzt, gibt min.insync.replicas die Mindestanzahl von Replikaten an, die einen Schreibvorgang bestätigen müssen, damit der Schreibvorgang als erfolgreich angesehen wird. Wenn dieses Minimum nicht erreicht werden kann, löst der Hersteller eine Ausnahme aus (entweder oder). NotEnoughReplicas NotEnoughReplicasAfterAppend Sie können die Werte in min.insync.replicas und acks zusammen verwenden, um langfrist igere Beständigkeitsgarantien durchzuse tzen. Zum Beispiel könnten Sie ein Thema mit dem Replikationsfaktor 3 erstellen, min.insyn c.replicas auf 2 einstellen und mit acks von `"all"` produzieren. Dadurch wird sicherges tellt, dass der Produzent eine Ausnahme auslöst, wenn die Mehrheit der Replikate keinen Schreibvorgang erhält.
num.io.threads	Die Anzahl der Threads, die der Server für die Verarbeitung von Anforderungen verwendet, einschließlich Datenträger-E/A.
num.network.threads	Die Anzahl der Threads, die der Server zum Empfangen von Anfragen aus dem Netzwerk und zum Senden von Antworten verwendet.
num.partitions	Standardanzahl der Protokollpartitionen pro Thema.
num.recovery.threads.per.data.dir	Die Anzahl der Threads pro Datenverzeichnis, die für die Protokollwiederherstellung beim Startup und zum Bereinigen beim Herunterf ahren verwendet werden sollen.

Name	Beschreibung
num.replica.fetchers	Die Anzahl der Abfrage-Threads, die zum Replizieren von Nachrichten von einem Quell-Broker verwendet werden. Wenn Sie diesen Wert erhöhen, können Sie den Grad der I/O-Parallelität im Follower-Broker erhöhen.
offsets.retention.minutes	Nachdem eine Konsumentengruppe alle Konsumenten verliert (d. h. sie ist dann leer), werden die Offsets für diesen Aufbewahr ungszeitraum aufbewahrt, bevor sie verworfen werden. Bei eigenständigen Verbrauchern (d. h. diejenige, die manueller Zuweisung verwenden) sind Offsets nach dem Zeitpunkt des letzten Commits zusätzlich dieser Aufbewahrungsfrist abgelaufen.
offsets.topic.replication.factor	Der Replikationsfaktor für das Offsets-T hema. Setzen Sie diesen Wert höher, um die Verfügbarkeit sicherzustellen. Die interne Themenerstellung schlägt fehl, bis die Cluster-Größe diese Anforderung des Replikati onsfaktors erfüllt.
replica.fetch.max.bytes	Anzahl der Bytes von Nachrichten, die für jede Partition abgerufen werden sollen. Es handelt sich nicht um ein absolutes Maximum. Wenn der erste Datensatz-Batch in der ersten nicht leeren Partition des Abrufs größer ist als dieser Wert, wird der Datensatz-Batch zurückgeg eben, damit Fortschritte gemacht werden können. Die Eigenschaften message.max.bytes (Broker-Konfiguration) oder max.messa ge.bytes (Themenkonfiguration) geben die maximale vom Broker akzeptierte Datensatz-Batch-Größe an.

Name	Beschreibung
replica.fetch.response.max.bytes	Die maximale Anzahl von Bytes, die für die gesamte Abrufantwort erwartet wird. Datensätze werden in Batches abgerufen und wenn der erste Datensatz-Batch in der ersten nicht leeren Partition des Abrufs größer ist als dieser Wert, wird der Datensatz-Batch weiterhin zurückgegeben, damit Fortschritte gemacht werden können. Es handelt sich nicht um ein absolutes Maximum. Die Eigenschaften message.max.bytes (Broker-Konfiguration) oder max.message.bytes (Themenkonfiguration) geben die maximale vom Broker akzeptierte Datensatzstapelgröße an.
replica.lag.time.max.ms	Wenn ein Follower für mindestens diese Anzahl von Millisekunden keine Abrufanforderungen gesendet hat oder nicht bis zum Protokollendversatz des Leaders konsumiert hat, entfernt der Leader den Follower aus dem ISR. MinValue: 10000 MaxValue = 30000

Name	Beschreibung
replica.selector.class	Der vollqualifizierte Klassenname, der implementiert wird. ReplicaSelector Der Broker verwendet diesen Wert, um das bevorzugte Lesereplikat zu finden. Wenn Sie Apache Kafka Version 2.4.1 oder höher verwenden und es Verbrauchern erlauben möchten, vom nächstgelegenen Replikat abzurufen, setzen Sie diese Eigenschaft auf `org.apache.kafka.common.replica.RackAwareReplicaSelector`. Weitere Informationen finden Sie unter the section called “Apache Kafka Version 2.4.1 (verwenden Sie stattdessen 2.4.1.1)”.
replica.socket.receive.buffer.bytes	Der Socket-Empfangspuffer für Netzwerkanforderungen.
socket.receive.buffer.bytes	Der SO_RCVBUF-Puffer der Socket-Server-Sockets. Der Mindestwert, den Sie für diese Eigenschaft festlegen können, ist -1. Wenn der Wert -1 ist, verwendet Amazon MSK den Betriebssystemstandard.
socket.request.max.bytes	Die maximale Anzahl von Bytes in einer Socket-Anfrage.
socket.send.buffer.bytes	Der SO_SNDBUF-Puffer der Socket-Server-Sockets. Der Mindestwert, den Sie für diese Eigenschaft festlegen können, ist -1. Wenn der Wert -1 ist, verwendet Amazon MSK den Betriebssystemstandard.

Name	Beschreibung
transaction.max.timeout.ms	Maximales Timeout für Transaktionen. Wenn die angeforderte Transaktionszeit eines Clients diesen Wert überschreitet, gibt der Broker einen Fehler in InitProducerIdRequest zurück. So wird ein zu großer Timeout auf Client-Se ite verhindert, der Verbraucher am Lesen aus Themen, die in der Transaktion vorhanden sind, hindern könnte.
transaction.state.log.min.isr	Überschriebene min.insync.replicas-Konfigu ration für das Transaktionsthema.
transaction.state.log.replication.factor	Der Replikationsfaktor für das Transakti onsthema. Setzen Sie diese Eigenschaft auf einen höheren Wert, um die Verfügbarkeit zu erhöhen. Die interne Themenerstellung schlägt fehl, bis die Cluster-Größe diese Anforderung des Replikationsfaktors erfüllt.
transactional.id.expiration.ms	Die Zeit in Millisekunden, in der der Transakti onskoordinator auf Aktualisierungen des Transaktionsstatus für die aktuelle Transaktion wartet, bevor der Koordinator seine Transakti ons-ID ablaufen lässt. Diese Einstellung beeinflusst auch den Ablauf der Produzent en-ID, da sie bewirkt, dass die Produzent en-IDs ablaufen, wenn diese Zeit nach dem letzten Schreibvorgang mit der angegebenen Produzenten-ID verstrichen ist. Produzenten-IDs laufen aufgrund der Aufbewahrungseinst ellungen für das Thema möglicherweise früher ab, wenn der letzte Schreibvorgang aus der Produzenten-ID gelöscht wird. Der Mindestwert für diese Eigenschaft ist 1 Millisekunde.

Name	Beschreibung
unclean.leader.election.enable	Gibt an, ob Replikate, die nicht im ISR-Satz enthalten sind, als letztes Mittel als Führer dienen sollen, auch wenn dies zu Datenverlust führen kann.
zookeeper.connection.timeout.ms	ZooKeeper Modus-Cluster. Maximale Zeit, bis zu der der Client wartet, um eine Verbindung herzustellen. ZooKeeper Wenn Sie diesen Wert nicht festlegen, wird der Wert in zookeeper .session.timeout.ms verwendet. MinValue = 6000 MaxValue (einschließlich) = 18000
zookeeper.session.timeout.ms	ZooKeeper mehr Cluster. Das Zeitlimit für die ZooKeeper Apache-Sitzung in Millisekunden. MinValue = 6000 MaxValue (einschließlich) = 18000

Weitere Informationen dazu, wie Sie eine benutzerdefinierte MSK-Konfiguration erstellen, alle Konfigurationen auflisten oder diese beschreiben können, finden Sie unter the section called “Konfigurationsvorgänge”. Informationen zum Erstellen eines MSK-Clusters mit einer benutzerdefinierten MSK-Konfiguration oder zum Aktualisieren eines Clusters mit einer neuen benutzerdefinierten Konfiguration finden Sie unter Funktionsweise.

Wenn Sie den vorhandenen MSK-Cluster mit einer benutzerdefinierten MSK-Konfiguration aktualisieren, führt Amazon MSK bei Bedarf unter Verwendung bewährter Methoden fortlaufende Neustarts durch, um Ausfallzeiten für Kunden zu minimieren. Nachdem Amazon MSK jeden Broker neu gestartet hat, warten Amazon MSK, bis der Broker Daten verarbeitet hat, die während des Konfigurations-Updates möglicherweise verpasst wurden, bevor zum nächsten Broker übergegangen wird.

Dynamische Konfiguration

Zusätzlich zu den Konfigurationseigenschaften, die Amazon MSK bereitstellt, können Sie Konfigurationseigenschaften, für die kein Broker-Neustart erforderlich ist, auf Cluster- und Broker-Ebene dynamisch festlegen. Sie können einige Konfigurationseigenschaften dynamisch festlegen. Dies sind die Eigenschaften, die in der Tabelle unter Broker-Konfigurationen in der Apache-Kafka-Dokumentation nicht als schreibgeschützt markiert sind. Informationen zur dynamischen Konfiguration und zu Beispielbefehlen finden Sie unter Aktualisieren der Broker-Konfigurationen in der Apache-Kafka-Dokumentation.

> **Note**
>
> Sie können die Eigenschaft `advertised.listeners` festlegen, die Eigenschaft `listeners` hingegen nicht.

Konfiguration auf Themenebene

Sie können Apache Kafka-Befehle verwenden, um Konfigurationseigenschaften auf Themenebene für neue und vorhandene Themen festzulegen oder zu ändern. Weitere Informationen zu Konfigurationseigenschaften auf Themenebene und Beispiele zum Festlegen dieser Eigenschaften finden Sie unter Konfigurationen auf Themenebene in der Apache-Kafka-Dokumentation.

Status der Konfiguration

Eine Amazon-MSK-Konfiguration kann sich in einem der folgenden Status befinden. Um einen Vorgang an einer Konfiguration durchzuführen, muss sich die Konfiguration im Status `ACTIVE` oder `DELETE_FAILED` befinden:

- `ACTIVE`
- `DELETING`
- `DELETE_FAILED`

Die Standardkonfiguration von Amazon MSK

Wenn Sie einen MSK-Cluster erstellen, ohne eine benutzerdefinierte MSK-Konfiguration anzugeben, erstellt und verwendet Amazon MSK eine Standardkonfiguration mit den in der folgenden Tabelle

angegebenen Werten. Bei Eigenschaften, die nicht in dieser Tabelle enthalten sind, verwendet Amazon MSK die Standardwerte, die Ihrer Version von Apache Kafka zugeordnet sind. Eine Liste dieser Standardwerte finden Sie unter Apache Kafka Configuration.

Standardkonfigurationswerte

Name	Beschreibung	Standardwert für Cluster mit nicht-gestaffeltem Speicher	Standardwert für Cluster mit aktivierter gestaffelter Speicherung
allow.everyone.if.no.acl.found	Wenn keine Ressourcenmuster mit einer bestimmten Ressource übereinstimmen, sind der Ressource keine Zugriffskontrolllisten zugeordnet. Wenn diese Eigenschaft auf `true` gesetzt ist, kann jeder auf die Ressource zugreifen, nicht nur die Superuser.	`true`	`true`
auto.create.topics.enable	Aktiviert die automatische Erstellung eines Themas auf dem Server.	`false`	`false`
auto.leader.rebalance.enable	Aktiviert den automatischen Führungsausgleich. Ein Hintergrund-Thread prüft den Führungsausgleich und löst, wenn	`true`	`true`

Name	Beschreibung	Standardwert für Cluster mit nicht-gestaffeltem Speicher	Standardwert für Cluster mit aktivierter gestaffelter Speicherung
	erforderlich, diesen in regelmäßigen Abständen aus.		
default.replication.factor	Standardreplikationsfaktoren für automatisch erstellte Themen.	3 für Cluster in 3 Availability Zones und 2 für Cluster in 2 Availability Zones.	3 für Cluster in 3 Availability Zones und 2 für Cluster in 2 Availability Zones.

Name	Beschreibung	Standardwert für Cluster mit nicht-gestaffeltem Speicher	Standardwert für Cluster mit aktivierter gestaffelter Speicherung
local.retention.bytes	Die maximale Größe der lokalen Protokollsegmente für eine Partition, bevor die alten Segmente gelöscht werden. Wenn Sie diesen Wert nicht festlegen, wird der Wert in log.retention.bytes verwendet. Der effektive Wert sollte immer kleiner oder gleich dem Wert log.retention.bytes sein. Ein Standardwert von -2 bedeutet, dass kein Grenzwert für die lokale Aufbewahrung vorhanden ist. Dies entspricht der retention.ms/bytes-Einstellung von -1. Die Eigenschaften local.retention.ms und local.retention.bytes ähneln log.retention, da sie verwendet werden, um zu bestimmen, wie lange die Protokollsegmente	-2 für unbegrenzt	-2 für unbegrenzt

Name	Beschreibung	Standardwert für Cluster mit nicht-gestaffeltem Speicher	Standardwert für Cluster mit aktivierter gestaffelter Speicherung
	im lokalen Speicher verbleiben sollen. Bestehende log.retention.*-Konfigurationen sind Aufbewahrungskonfigurationen für die Themenpartition. Dies umfasst sowohl lokalen als auch Remote-Speicher. Gültige Werte: Ganzzahlen in [-2; +Inf]		

Name	Beschreibung	Standardwert für Cluster mit nicht-gestaffeltem Speicher	Standardwert für Cluster mit aktivierter gestaffelter Speicherung
local.retention.ms	Die Anzahl der Millisekunden, die das lokale Protokollsegment vor dem Löschen beibehalten werden soll. Wenn Sie diesen Wert nicht festlegen, verwendet Amazon MSK den Wert in log.retention.ms. Der effektive Wert sollte immer kleiner oder gleich dem Wert log.retention.bytes sein. Ein Standardwert von -2 bedeutet, dass kein Grenzwert für die lokale Aufbewahrung vorhanden ist. Dies entspricht der retention.ms/bytes-Einstellung von -1. Die Werte local.retention.ms und local.retention.bytes ähneln log.retention. MSK verwendet diese Konfiguration, um zu bestimmen, wie lange die Protokollsegmente	-2 für unbegrenzt	-2 für unbegrenzt

Name	Beschreibung	Standardwert für Cluster mit nicht-gestaffeltem Speicher	Standardwert für Cluster mit aktivierter gestaffelter Speicherung
	im lokalen Speicher verbleiben sollen. Bestehende log.retention.*-Konfigurationen sind Aufbewahrungskonfigurationen für die Themenpartition. Dies umfasst sowohl lokalen als auch Remote-Speicher. Gültige Werte sind Ganzzahlen größer als 0.		

Name	Beschreibung	Standardwert für Cluster mit nicht-gestaffeltem Speicher	Standardwert für Cluster mit aktivierter gestaffelter Speicherung
log.message.timestamp.difference.max.ms	Die maximal zulässige Diskrepanz zwischen dem Zeitstempel beim Empfang einer Nachricht durch den Broker und dem in der Nachricht angegebenen Zeitstempel. Bei log.message.timestamp.type= wird eine Nachricht zurückgewiesenCreateTime, wenn der Unterschied im Zeitstempel diesen Schwellenwert überschreitet. Diese Konfiguration wird LogAppend ignoriert, wenn log.message.timestamp.type= Zeit. Der maximal zulässige Zeitstempelunterschied sollte nicht größer als log.retention.ms sein, um unnötig häufiges Protokoll-Rolling zu vermeiden.	9223372036854775807	86400000 für Kafka 2.8.2.tiered

Name	Beschreibung	Standardwert für Cluster mit nicht-gestaffeltem Speicher	Standardwert für Cluster mit aktivierter gestaffelter Speicherung
log.segment.bytes	Die maximale Größe einer einzelnen Protokolldatei.	1073741824	134217728

Name	Beschreibung	Standardwert für Cluster mit nicht-gestaffeltem Speicher	Standardwert für Cluster mit aktivierter gestaffelter Speicherung
min.insync.replicas	Wenn ein Produzent den Wert von acks (Bestätigung, die der Produzent vom Kafka-Brocker erhält) auf `"all"` (oder `"-1"`) setzt, gibt der Wert in min.insync.replicas die Mindestanzahl von Replikaten an, die einen Schreibvorgang bestätigen müssen, damit der Schreibvorgang als erfolgreich angesehen wird. Wenn dieser Wert dieses Minimum nicht erreicht, löst der Producer eine Ausnahme aus (entweder oder). NotEnoughReplicas NotEnoughReplicasAfterAppend Wenn Sie die Werte in min.insync.replicas und acks zusammen verwenden, können Sie langfristigere Beständigkeitsgarantien durchsetzen.	2 für Cluster in 3 Availability Zones und 1 für Cluster in 2 Availability Zones.	2 für Cluster in 3 Availability Zones und 1 für Cluster in 2 Availability Zones.

Name	Beschreibung	Standardwert für Cluster mit nicht-gestaffeltem Speicher	Standardwert für Cluster mit aktivierter gestaffelter Speicherung
	Zum Beispiel könnten Sie ein Thema mit dem Replikationsfaktor 3 erstellen, min.insync.replicas auf 2 einstellen und mit acks von `"all"` produzieren. Dadurch wird sichergestellt, dass der Produzent eine Ausnahme auslöst, wenn die Mehrheit der Replikate keinen Schreibvorgang erhält.		
num.io.threads	Anzahl der Threads, die der Server für die Erzeugung von Anfragen verwendet, eventuell einschließlich Datenträger-I/O.	8	max (8, vCPUs), wobei vCPUs von der Instance-Größe des Brokers abhängen
num.network.threads	Anzahl der Threads, die der Server zum Empfangen von Anfragen aus dem Netzwerk und zum Senden von Antworten an das Netzwerk verwendet.	5	max (5, vCPUs) / 2, wobei vCPUs von der Instance-Größe des Brokers abhängen

Name	Beschreibung	Standardwert für Cluster mit nicht-gestaffeltem Speicher	Standardwert für Cluster mit aktivierter gestaffelter Speicherung
num.partitions	Standardanzahl der Protokollpartitionen pro Thema.	1	1
num.replica.fetchers	Anzahl der Abfrage-Threads, die zum Replizieren von Nachrichten von einem Quell-Broker verwendet werden. Wenn Sie diesen Wert erhöhen, können Sie den Grad der I/O-Parallelität im Follower-Broker erhöhen.	2	max (2, vCPUs / 4) wobei vCPUs von der Instance-Größe des Brokers abhängen
remote.log.msk.disable.policy	Wird zusammen mit remote.storage.enable verwendet, um die gestaffelte Speicherung zu deaktivieren. Setzen Sie diese Richtlinie auf Löschen, um anzugeben, dass Daten im gestaffelten Speicher gelöscht werden, wenn Sie remote.storage.enable auf Falsch setzen.	N/A	DELETE

Name	Beschreibung	Standardwert für Cluster mit nicht-gestaffeltem Speicher	Standardwert für Cluster mit aktivierter gestaffelter Speicherung
remote.log.reader.threads	Größe des Threadpools für den Remote-Protokollleser, der bei der Planung von Aufgaben zum Abrufen von Daten aus dem Remote-Speicher verwendet wird.	N/A	max (10, vCPUs * 0,67), wobei vCPUs von der Instance-Größe des Brokers abhängen

Name	Beschreibung	Standardwert für Cluster mit nicht-gestaffeltem Speicher	Standardwert für Cluster mit aktivierter gestaffelter Speicherung
remote.storage.enable	Aktiviert gestaffelte (Remote-)Speicherung für ein Thema, wenn dieser Wert auf Wahr gesetzt ist. Deaktiviert die gestaffelte Speicherung auf Themenebene, wenn der Wert auf Falsch gesetzt ist und remote.log.msk.disable.policy auf Löschen gesetzt ist. Wenn Sie die gestaffelte Speicherung deaktivieren, löschen Sie Daten aus dem Remote-Speicher. Wenn Sie die gestaffelte Speicherung für ein Thema deaktiviert haben, können Sie sie nicht erneut aktivieren.	false	true

Name	Beschreibung	Standardwert für Cluster mit nicht-gestaffeltem Speicher	Standardwert für Cluster mit aktivierter gestaffelter Speicherung
replica.lag.time.max.ms	Wenn ein Follower für mindestens diese Anzahl von Millisekunden keine Abrufanforderungen gesendet hat oder nicht bis zum Protokollendversatz des Leaders konsumiert hat, entfernt der Leader den Follower aus dem ISR.	30000	30000

Name	Beschreibung	Standardwert für Cluster mit nicht-gestaffeltem Speicher	Standardwert für Cluster mit aktivierter gestaffelter Speicherung
retention.ms	Plichtfeld. Die Mindestzeit beträgt 3 Tage. Es gibt keine Standardeinstellung, da die Einstellung ein Pflichtfeld ist. Amazon MSK verwendet den Wert retention.ms zusammen mit local.retention.ms, um zu bestimmen, wann Daten vom lokalen zum gestaffelten Speicher verschoben werden. Der Wert local.retention.ms gibt an, wann Daten vom lokalen in den gestaffelten Speicher verschoben werden sollen. Der Wert retention.ms gibt an, wann Daten aus dem gestaffelten Speicher entfernt (d. h. aus dem Cluster entfernt) werden sollen. Gültige Werte: Ganzzahlen in [-1; +Inf]	Mindestens 259 200 000 Millisekunden (3 Tage). -1 für unendliche Aufbewahrung.	Mindestens 259 200 000 Millisekunden (3 Tage). -1 für unendliche Aufbewahrung.

Name	Beschreibung	Standardwert für Cluster mit nicht-gestaffeltem Speicher	Standardwert für Cluster mit aktivierter gestaffelter Speicherung
socket.receive.buffer.bytes	Der SO_RCVBUF-Puffer der Socket-Server-Sockets. Wenn der Wert -1 ist, wird der Standardwert des Betriebssystems verwendet.	102400	102400
socket.request.max.bytes	Maximale Anzahl von Bytes in einer Socket-Anforderung.	104857600	104857600
socket.send.buffer.bytes	Der SO_SNDBUF-Puffer der Socket-Server-Sockets. Wenn der Wert -1 ist, wird der Standardwert des Betriebssystems verwendet.	102400	102400
unclean.leader.election.enable	Gibt an, ob Replikate, die nicht in der ISR-Gruppe enthalten sind, als letztes Mittel als Führer dienen sollen, auch wenn dies zu Datenverlust führen kann.	true	false

Name	Beschreibung	Standardwert für Cluster mit nicht-gestaffeltem Speicher	Standardwert für Cluster mit aktivierter gestaffelter Speicherung
zookeeper.session.timeout.ms	Das Zeitlimit für die ZooKeeper Apache-Sitzung in Millisekunden.	18000	18000
zookeeper.set.acl	Legen Sie den Client fest, um sichere Zugriffssteuerungslisten zu verwenden.	false	false

Weitere Informationen zum Festlegen von benutzerdefinierten Konfigurationswerten finden Sie unter the section called “Benutzerdefinierte -Konfigurationen”.

Richtlinien für die Konfiguration der gestaffelten Speicherung auf Themenebene

Im Folgenden finden Sie Standardeinstellungen und Einschränkungen bei der Konfiguration der gestaffelten Speicherung auf Themenebene.

- Amazon MSK unterstützt keine kleineren Protokollsegmentgrößen für Themen, für die gestaffelte Speicherung aktiviert ist. Wenn Sie ein Segment erstellen möchten, gibt es eine Mindestgröße für das Protokoll-Segment von 48 MiB oder eine Mindest-Segment-Rollzeit von 10 Minuten. Diese Werte sind den Eigenschaften segment.bytes und segment.ms zugeordnet.
- Der Wert von local.retention.ms/bytes darf dem Wert von retention.ms/bytes nicht entsprechen oder diesen überschreiten. Dies ist die Aufbewahrungseinstellung der gestaffelten Speicherung.
- Der Standardwert für local.retention.ms/bytes ist -2. Das bedeutet, dass der Wert retention.ms für local.retention.ms/bytes verwendet wird. In diesem Fall verbleiben die Daten sowohl im lokalen als auch im gestaffelten Speicher (jeweils eine Kopie), und sie laufen zusammen ab. Bei dieser Option wird eine Kopie der lokalen Daten dauerhaft im Remote-Speicher gespeichert. In diesem Fall stammen die aus dem Verbraucherdatenverkehr gelesenen Daten aus dem lokalen Speicher.

- Der Standardwert für retention.ms ist 7 Tage. Es gibt keine Standard-Größenbeschränkung für retention.bytes.
- Der Mindestwert für retention.ms/bytes ist -1. Dies bedeutet unendliche Aufbewahrung.
- Der Mindestwert für local.retention.ms/bytes ist -2. Dies bedeutet unendliche Aufbewahrung für den lokalen Speicher. Dies entspricht der Einstellung retention.ms/bytes auf -1.
- Die Konfiguration retention.ms auf Themenebene ist für Themen mit aktivierter gestaffelter Speicherung obligatorisch. Der Mindestwert für retention.ms ist 3 Tage.

Amazon-MSK-Konfigurationsvorgänge

In diesem Thema wird beschrieben, wie benutzerdefinierte MSK-Konfigurationen erstellt und Vorgänge an diesen ausgeführt werden. Informationen zur Verwendung von MSK-Konfigurationen zum Erstellen oder Aktualisieren von Clustern finden Sie unter Funktionsweise.

Dieses Thema enthält die folgenden Abschnitte:

So erstellen Sie eine MSK-Konfiguration

1. Erstellen Sie eine Datei, in der Sie die festzulegenden Konfigurationseigenschaften und die Werte angeben, die Sie ihnen zuweisen möchten. Im Folgenden finden Sie den Inhalt einer Beispielkonfigurationsdatei.

   ```
   auto.create.topics.enable = true

   log.roll.ms = 604800000
   ```

2. Führen Sie den folgenden AWS CLI Befehl aus und ersetzen Sie *config-file-path durch den Pfad* zu der Datei, in der Sie Ihre Konfiguration im vorherigen Schritt gespeichert haben.

> **Note**
>
> Der Name, den Sie für Ihre Konfiguration auswählen, muss mit dem folgenden regulären Ausdruck übereinstimmen: „^[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
```

Im Folgenden finden Sie ein Beispiel für eine erfolgreiche Antwort nach der Ausführung dieses Befehls.

```
{
    "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. Der vorherige Befehl gibt einen Amazon-Ressourcennamen (ARN) für die neue Konfiguration zurück. Speichern Sie diesen ARN, da Sie bei anderen Befehlen auf diese Konfiguration verweisen müssen. Wenn Sie den Konfigurations-ARN verlieren, finden Sie ihn in der Konfigurationsliste in Ihrem Konto wieder.

So aktualisieren Sie eine MSK-Konfiguration

1. Erstellen Sie eine Datei, in der Sie die zu aktualisierenden Konfigurationseigenschaften angeben, und die Werte, die Sie ihnen zuweisen möchten. Im Folgenden finden Sie den Inhalt einer Beispielkonfigurationsdatei.

```
auto.create.topics.enable = true
```

```
min.insync.replicas = 2
```

2. Führen Sie den folgenden AWS CLI -Befehl aus und ersetzen Sie *config-file-path* durch den Pfad der Datei, in der Sie die Konfiguration im vorherigen Schritt gespeichert haben.

   Ersetzen Sie *configuration-arn* durch den ARN, den Sie beim Erstellen der Konfiguration erhalten haben. Wenn Sie den ARN beim Erstellen der Konfiguration nicht gespeichert haben, können Sie den `list-configurations`-Befehl verwenden, um alle Konfigurationen in Ihrem Konto aufzulisten. Die Konfiguration, die Sie in der Liste haben möchten, wird in der Antwort angezeigt. Der ARN der Konfiguration wird ebenfalls in dieser Liste angezeigt.

```
aws kafka update-configuration --arn configuration-arn --description "Example
 configuration revision description." --server-properties fileb://config-file-path
```

3. Im Folgenden finden Sie ein Beispiel für eine erfolgreiche Antwort nach der Ausführung dieses Befehls.

```
{
    "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
    }
}
```

So löschen Sie eine MSK-Konfiguration

Das folgende Verfahren zeigt, wie Sie eine Konfiguration löschen, die nicht einem Cluster angefügt ist. Sie können eine Konfiguration nicht löschen, die einem Cluster angefügt ist.

1. Um dieses Beispiel auszuführen, ersetzen Sie *configuration-arn* durch den ARN, den Sie beim Erstellen der Konfiguration erhalten haben. Wenn Sie den ARN beim Erstellen der Konfiguration nicht gespeichert haben, können Sie den `list-configurations`-Befehl verwenden, um alle Konfigurationen in Ihrem Konto aufzulisten. Die Konfiguration, die Sie in der Liste haben möchten, wird in der Antwort angezeigt. Der ARN der Konfiguration wird ebenfalls in dieser Liste angezeigt.

```
aws kafka delete-configuration --arn configuration-arn
```

2. Im Folgenden finden Sie ein Beispiel für eine erfolgreiche Antwort nach der Ausführung dieses Befehls.

```
{
    "arn": " arn:aws:kafka:us-east-1:123456789012:configuration/SomeTest/
abcdabcd-1234-abcd-1234-abcd123e8e8e-1",
    "state": "DELETING"
}
```

So beschreiben Sie eine MSK-Konfiguration

1. Der folgende Befehl gibt Metadaten zur Konfiguration zurück. Um eine detaillierte Beschreibung der Konfiguration zu erhalten, führen Sie `describe-configuration-revision` aus .

   Um dieses Beispiel auszuführen, ersetzen Sie *configuration-arn* durch den ARN, den Sie beim Erstellen der Konfiguration erhalten haben. Wenn Sie den ARN beim Erstellen der Konfiguration nicht gespeichert haben, können Sie den `list-configurations`-Befehl verwenden, um alle Konfigurationen in Ihrem Konto aufzulisten. Die Konfiguration, die Sie in der Liste haben möchten, wird in der Antwort angezeigt. Der ARN der Konfiguration wird ebenfalls in dieser Liste angezeigt.

```
aws kafka describe-configuration --arn configuration-arn
```

2. Im Folgenden finden Sie ein Beispiel für eine erfolgreiche Antwort nach der Ausführung dieses Befehls.

```
{
    "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"
}
```

So beschreiben Sie eine MSK-Konfigurationsversion

Wenn Sie den `describe-configuration`-Befehl verwenden, um eine MSK-Konfiguration zu beschreiben, erhalten Sie die Metadaten der Konfiguration. Um eine Beschreibung der Konfiguration zu erhalten, verwenden Sie den Befehl `describe-configuration-revision`.

- Führen Sie den folgenden Befehl aus und ersetzen Sie *`configuration-arn`* durch den ARN, den Sie beim Erstellen der Konfiguration erhalten haben. Wenn Sie den ARN beim Erstellen der Konfiguration nicht gespeichert haben, können Sie den `list-configurations`-Befehl verwenden, um alle Konfigurationen in Ihrem Konto aufzulisten. Die Konfiguration, die Sie in der Liste suchen, wird in der Antwort angezeigt. Der ARN der Konfiguration wird ebenfalls in dieser Liste angezeigt.

  ```
  aws kafka describe-configuration-revision --arn configuration-arn --revision 1
  ```

  Im Folgenden finden Sie ein Beispiel für eine erfolgreiche Antwort nach der Ausführung dieses Befehls.

  ```
  {
      "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
  }
  ```

  Der Wert von `ServerProperties` wird mit base64 codiert. Wenn Sie einen base64-Decoder (z. B. https://www.base64decode.org/) verwenden, um den Wert manuell zu dekodieren, erhalten Sie den Inhalt der ursprünglichen Konfigurationsdatei, mit der Sie die benutzerdefinierte Konfiguration erstellt haben. In diesem Fall erhalten Sie Folgendes:

```
auto.create.topics.enable = true

log.roll.ms = 604800000
```

So listen Sie alle MSK-Konfigurationen in Ihrem Konto für die aktuelle Region auf

- Führen Sie den folgenden Befehl aus.

  ```
  aws kafka list-configurations
  ```

  Im Folgenden finden Sie ein Beispiel für eine erfolgreiche Antwort nach der Ausführung dieses Befehls.

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

MSK Serverless

> **Note**
>
> MSK Serverless ist in den Regionen USA Ost (Nord-Virginia), USA Ost (Ohio), USA West (Oregon), Kanada (Zentral), Asien-Pazifik (Mumbai), Asien-Pazifik (Seoul), Asien-Pazifik (Singapur), Asien-Pazifik (Sydney), Asien-Pazifik (Tokio), Europa (Frankfurt), Europa (Stockholm) Europa (Irland), Europa (London) und Europa (Paris) verfügbar.

MSK Serverless ist ein Cluster-Typ für Amazon MSK, mit dem Sie Apache Kafka ausführen können, ohne die Cluster-Kapazität verwalten und skalieren zu müssen. Die Kapazität wird automatisch bereitgestellt und skaliert, während gleichzeitig die Partitionen in Ihrem Thema verwaltet werden, sodass Sie Daten streamen können, ohne über die richtige Größe oder Skalierung von Clustern nachdenken zu müssen. MSK Serverless bietet ein durchsatzbasiertes Preismodell. Sie zahlen nur für das, was Sie tatsächlich nutzen. Erwägen Sie die Verwendung eines Serverless-Clusters, wenn Ihre Anwendungen On-Demand-Streaming-Kapazität benötigen, die automatisch hoch- und herunterskaliert wird.

MSK Serverless ist vollständig mit Apache Kafka kompatibel, sodass Sie beliebige kompatible Client-Anwendungen zur Erzeugung und Nutzung von Daten verwenden können. Es kann auch in folgende Services integriert werden:

- AWS PrivateLink um private Konnektivität bereitzustellen
- AWS Identity and Access Management (IAM) für die Authentifizierung und Autorisierung mit Java- und Nicht-Java-Sprachen. Anweisungen zur Konfiguration von Clients für IAM finden Sie unter Konfiguration von Clients für die IAM-Zugriffssteuerung.
- AWS Glue Schema Registry für die Schemaverwaltung
- Amazon Managed Service für Apache Flink für Apache-Flink-basierte Stream-Verarbeitung
- AWS Lambda für die Verarbeitung von Ereignissen

> **Note**
>
> MSK Serverless erfordert IAM-Zugriffssteuerung für alle Cluster. Apache-Kafka-Zugriffssteuerungslisten (ACLs) werden nicht unterstützt. Weitere Informationen finden Sie unter the section called “IAM-Zugriffssteuerung”.

Informationen zu Servicekontingenten, die für MSK Serverless gelten, finden Sie unter the section called “Kontingent für Serverless-Cluster”.

Im Folgenden finden Sie Informationen zu den ersten Schritten mit Serverless-Clustern und erfahren Sie mehr über die Konfigurations- und Überwachungsoptionen für Serverless-Cluster.

Themen

- Erste Schritte mit MSK-Serverless-Clustern
- Konfiguration für Serverless-Cluster
- Überwachen von Serverless-Clustern

Erste Schritte mit MSK-Serverless-Clustern

Dieses Tutorial zeigt Ihnen ein Beispiel dafür, wie Sie einen MSK-Serverless-Cluster erstellen, einen Client-Computer erstellen, der darauf zugreifen kann, und den Client verwenden, um Themen auf dem Cluster zu erstellen und Daten in diese Themen zu schreiben. Dieses Beispiel zeigt nicht alle Optionen, die Sie auswählen können, wenn Sie einen Serverless-Cluster erstellen. In verschiedenen Teilen dieses Tutorials wählen wir aus Gründen der Einfachheit die Standardoptionen. Dies bedeutet nicht, dass dies die einzigen Optionen sind, die funktionieren, um einen Serverless-Cluster einzurichten. Sie können auch die AWS CLI oder die Amazon MSK-API verwenden. Weitere Informationen finden Sie in der Amazon-MSK-API-Referenz 2.0.

Themen

- Schritt 1: Einen MSK-Serverless-Cluster erstellen
- Schritt 2: Erstellen einer IAM-Rolle
- Schritt 3: Einen Client-Computer erstellen
- Schritt 4: Ein Apache-Kafka-Thema erstellen
- Schritt 5: Produzieren und Verbrauchen von Daten
- Schritt 6: Löschen von Ressourcen

Schritt 1: Einen MSK-Serverless-Cluster erstellen

In diesem Schritt führen Sie zwei Aufgaben aus. Zunächst erstellen Sie einen MSK-Serverless-Cluster mit Standardeinstellungen. Zweitens sammeln Sie Informationen über den Cluster. Diese

Informationen benötigen Sie in späteren Schritten, wenn Sie einen Client erstellen, der Daten an den Cluster senden kann.

So erstellen Sie einen Serverless-Cluster

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon MSK-Konsole unter https://console.aws.amazon.com/msk/home.
2. Wählen Sie Cluster erstellen.
3. Lassen Sie für die Erstellungsmethode die Option Schnellerstellung ausgewählt. Mit der Option Schnellerstellung können Sie einen Serverless-Cluster mit Standardeinstellungen erstellen.
4. Geben Sie für Cluster-Name einen beschreibenden Namen ein, z. B. **`msk-serverless-tutorial-cluster`**.
5. Wählen Sie für Allgemeine Cluster-Eigenschaften Serverless als Cluster-Typ. Verwenden Sie die Standardwerte für die übrigen allgemeinen Cluster-Eigenschaften.
6. Beachten Sie die Tabelle unter Alle Cluster-Einstellungen. In dieser Tabelle sind die Standardwerte für wichtige Einstellungen wie Netzwerk und Verfügbarkeit aufgeführt. Außerdem wird angegeben, ob Sie die einzelnen Einstellungen nach der Erstellung des Clusters ändern können. Um eine Einstellung zu ändern, bevor Sie den Cluster erstellen, sollten Sie unter Erstellungsmethode die Option Benutzerdefiniertes Erstellen auswählen.

   > **Note**
   >
   > Sie können Clients von bis zu fünf verschiedenen VPCs mit MSK-Serverless-Clustern verbinden. Damit Client-Anwendungen bei einem Ausfall in eine andere Availability Zone wechseln können, müssen Sie in jeder VPC mindestens zwei Subnetze angeben.

7. Wählen Sie Cluster erstellen.

So sammeln Sie Informationen über den Cluster

1. Wählen Sie im Abschnitt mit der Cluster-Zusammenfassung die Option Client-Informationen anzeigen. Diese Schaltfläche bleibt ausgegraut, bis Amazon MSK die Erstellung des Clusters abgeschlossen hat. Möglicherweise müssen Sie einige Minuten warten, bis die Schaltfläche aktiv wird, sodass Sie sie verwenden können.
2. Kopieren Sie die Zeichenfolge unter der Bezeichnung Endpunkt. Dies ist Ihre Bootstrap-Server-Zeichenfolge.

3. Wählen Sie die Registerkarte Eigenschaften aus.
4. Kopieren Sie im Abschnitt Netzwerkeinstellungen die IDs der Subnetze und der Sicherheitsgruppe und speichern Sie sie, da Sie diese Informationen später benötigen werden, um einen Client-Computer zu erstellen.
5. Wählen Sie eines der Subnetze aus. Dadurch wird die Amazon-VPC-Konsole geöffnet. Suchen Sie die ID der Amazon VPC, die dem Subnetz zugeordnet ist. Speichern Sie diese Amazon-VPC-ID zur späteren Verwendung.

Nächster Schritt

Schritt 2: Erstellen einer IAM-Rolle

Schritt 2: Erstellen einer IAM-Rolle

In diesem Schritt führen Sie zwei Aufgaben aus. Die erste Aufgabe besteht darin, eine IAM-Richtlinie zu erstellen, die Zugriff auf die Erstellung von Themen auf dem Cluster und das Senden von Daten an diese Themen gewährt. Die zweite Aufgabe besteht darin, eine IAM-Rolle zu erstellen und ihr diese Richtlinie zuzuordnen. In einem späteren Schritt erstellen wir einen Client-Computer, der diese Rolle übernimmt und sie verwendet, um ein Thema auf dem Cluster zu erstellen und Daten an dieses Thema zu senden.

So erstellen Sie eine IAM-Richtlinie, die es ermöglicht, Themen zu erstellen und in sie zu schreiben

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Richtlinien.
3. Wählen Sie Richtlinie erstellen aus.
4. Wählen Sie die Registerkarte JSON und ersetzen Sie dann den JSON-Code im Editor-Fenster durch den Folgenden.

   Ersetzen Sie *Region* durch den Code der AWS-Region , in der Sie Ihren Cluster erstellt haben. Ersetzen Sie *Konto-ID* durch Ihre Konto-ID. *msk-serverless-tutorial-cluster*Ersetzen Sie es durch den Namen Ihres serverlosen Clusters.

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

```
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:AlterCluster",
                "kafka-cluster:DescribeCluster"
            ],
            "Resource": [
                "arn:aws:kafka:region:Account-ID:cluster/msk-serverless-tutorial-
cluster/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:*Topic*",
                "kafka-cluster:WriteData",
                "kafka-cluster:ReadData"
            ],
            "Resource": [
                "arn:aws:kafka:region:Account-ID:topic/msk-serverless-tutorial-
cluster/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeGroup"
            ],
            "Resource": [
                "arn:aws:kafka:region:Account-ID:group/msk-serverless-tutorial-
cluster/*"
            ]
        }
    ]
}
```

Anleitungen zum Verfassen von sicheren Richtlinien finden Sie unter the section called “IAM-Zugriffssteuerung”.

5. Wählen Sie Next: Tags (Weiter: Tags) aus.
6. Klicken Sie auf Weiter: Prüfen.
7. Geben für den Richtliniennamen einen beschreibenden Namen ein, z. B. **msk-serverless-tutorial-policy**.

8. Wählen Sie Richtlinie erstellen aus.

So erstellen Sie eine IAM-Rolle und fügen ihr die Richtlinie an

1. Wählen Sie im Navigationsbereich Rollen.
2. Wählen Sie Rolle erstellen.
3. Wählen Sie unter Häufige Anwendungsfälle die Option EC2 und dann Weiter: Berechtigungen.
4. Geben Sie in das Suchfeld den Namen der Richtlinie ein, die Sie zuvor für dieses Tutorial erstellt haben. Aktivieren Sie anschließend das Kontrollkästchen links neben der Richtlinie.
5. Wählen Sie Next: Tags (Weiter: Tags) aus.
6. Klicken Sie auf Weiter: Prüfen.
7. Geben Sie für den Rollennamen einen beschreibenden Namen ein, z. B. **`msk-serverless-tutorial-role`**.
8. Wählen Sie Rolle erstellen aus.

Nächster Schritt

Schritt 3: Einen Client-Computer erstellen

Schritt 3: Einen Client-Computer erstellen

In diesem Schritt führen Sie zwei Aufgaben aus. Die erste Aufgabe besteht darin, eine Amazon-EC2-Instance zu erstellen, die als Apache-Kafka-Client-Computer verwendet werden soll. Die zweite Aufgabe besteht darin, Java- und Apache-Kafka-Tools auf dem Computer zu installieren.

Erstellen eines Client-Computers

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie Launch Instance (Instance starten) aus.
3. Geben Sie einen beschreibenden Namen für Ihren Client-Computer ein, z. B. **`msk-serverless-tutorial-client`**
4. Lassen Sie Amazon Linux 2 AMI (HVM) – Kernel 5.10, SSD Volume Type als Amazon Machine Image (AMI)-Typ ausgewählt.
5. Lassen Sie den t2.micro-Instance-Typ ausgewählt.

6. Wählen Sie unter Schlüsselpaar (Login) die Option Neues Schlüsselpaar erstellen. Geben Sie **`MSKServerlessKeyPair`** für Schlüsselpaar-Name ein. Wählen Sie dann Schlüsselpaar herunterladen. Alternativ können Sie ein vorhandenes Schlüsselpaar verwenden.
7. Wählen Sie für Netzwerkeinstellungen die Option Bearbeiten aus.
8. Geben Sie unter VPC die ID der Virtual Private Cloud (VPC) für Ihren Serverless-Cluster ein. Dies ist die VPC, die auf dem Amazon-VPC-Service basiert und dessen ID Sie nach der Erstellung des Clusters gespeichert haben.
9. Wählen Sie für Subnetz das Subnetz aus, dessen ID Sie nach der Erstellung des Clusters gespeichert haben.
10. Wählen Sie unter Firewall (Sicherheitsgruppen) die Sicherheitsgruppe aus, die dem Cluster zugeordnet ist. Dieser Wert funktioniert, wenn diese Sicherheitsgruppe über eine eingehende Regel verfügt, die Datenverkehr von der Sicherheitsgruppe zu sich selbst zulässt. Mit einer solchen Regel können Mitglieder derselben Sicherheitsgruppe miteinander kommunizieren. Weitere Informationen finden Sie unter Sicherheitsgruppenregeln im Amazon-VPC-Benutzerhandbuch.
11. Erweitern Sie den Abschnitt Erweiterte Details und wählen Sie die IAM-Rolle aus, die Sie in Schritt 2: Erstellen einer IAM-Rolle erstellt haben.
12. Wählen Sie Launch (Starten) aus.
13. Wählen Sie im linken Navigationsbereich die Option Instances aus. Aktivieren Sie dann das Kontrollkästchen in der Zeile, die die neu erstellte Amazon-EC2-Instance darstellt. Ab diesem Zeitpunkt nennen wir diese Instance den Client-Computer.
14. Wählen Sie Verbinden und folgen Sie den Anweisungen, um eine Verbindung zum Client-Computer herzustellen.

So richten Sie die Apache-Kafka-Client-Tools auf dem Client-Computer ein

1. Installieren Sie Java auf dem Client-Computer, indem Sie den folgenden Befehl ausführen:

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

2. Führen Sie die folgenden Befehle aus, um die Apache-Kafka-Tools zu erhalten, die wir zum Erstellen von Themen und zum Senden von Daten benötigen:

```
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
```

3. Wechseln Sie zum Verzeichnis `kafka_2.12-2.8.1/libs` und führen Sie dann den folgenden Befehl aus, um die Amazon-MSK-IAM-JAR-Datei herunterzuladen. Das Amazon-MSK-IAM-JAR ermöglicht dem Client-Computer den Zugriff auf den Cluster.

```
wget https://github.com/aws/aws-msk-iam-auth/releases/download/v1.1.1/aws-msk-iam-auth-1.1.1-all.jar
```

4. Wechseln Sie zum Verzeichnis `kafka_2.12-2.8.1/bin`. Kopieren Sie die folgenden Eigenschaften-Einstellungen und fügen Sie sie in eine neue Datei ein. Benennen Sie die Datei `client.properties` und speichern Sie sie.

```
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
```

Nächster Schritt

Schritt 4: Ein Apache-Kafka-Thema erstellen

Schritt 4: Ein Apache-Kafka-Thema erstellen

In diesem Schritt verwenden Sie den zuvor erstellten Client-Computer, um ein Thema auf dem Serverless-Cluster zu erstellen.

So erstellen Sie ein Thema und schreiben Daten darin

1. Ersetzen Sie im folgenden `export`-Befehl *`my-endpoint`* durch die Bootstrap-Server-Zeichenfolge, die Sie nach der Erstellung des Clusters gespeichert haben. Wechseln Sie dann zum Verzeichnis `kafka_2.12-2.8.1/bin` auf dem Client-Computer und führen Sie den `export`-Befehl aus.

```
export BS=my-endpoint
```

2. Führen Sie den folgenden Befehl aus, um ein Thema mit dem Namen `msk-serverless-tutorial` zu erstellen.

```
<path-to-your-kafka-installation>/bin/kafka-topics.sh --bootstrap-server $BS
 --command-config client.properties --create --topic msk-serverless-tutorial --
partitions 6
```

Nächster Schritt

Schritt 5: Produzieren und Verbrauchen von Daten

Schritt 5: Produzieren und Verbrauchen von Daten

In diesem Schritt produzieren und verbrauchen Sie Daten mithilfe des Themas, das Sie im vorherigen Schritt erstellt haben.

Erstellen und Verbrauchen von Nachrichten

1. Führen Sie den folgenden Befehl aus, um einen Konsolenproduzenten zu erstellen.

   ```
   <path-to-your-kafka-installation>/bin/kafka-console-producer.sh --broker-list $BS
    --producer.config client.properties --topic msk-serverless-tutorial
   ```

2. Geben Sie eine beliebige Nachricht ein, und drücken Sie Enter (Eingabetaste). Wiederholen Sie diesen Schritt zwei- oder dreimal. Jedes Mal, wenn Sie eine Zeile eingeben und Eingabe drücken, wird diese Zeile als separate Nachricht an Ihren Apache-Kafka-Cluster gesendet.
3. Lassen Sie die Verbindung zum Client-Computer geöffnet und öffnen Sie dann eine zweite separate Verbindung zu diesem Computer in einem neuen Fenster.
4. Verwenden Sie Ihre zweite Verbindung zum Client-Computer, um mit dem folgenden Befehl einen Konsolen-Verbraucher zu erstellen. Ersetzen Sie *my-endpoint* durch die Bootstrap-Server-Zeichenfolge, die Sie nach der Erstellung des Clusters gespeichert haben.

   ```
   <path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
   server my-endpoint --consumer.config client.properties --topic msk-serverless-
   tutorial --from-beginning
   ```

   Sie sehen die Nachrichten, die Sie zuvor eingegeben haben, als Sie den Konsolenproduzentenbefehl verwendet haben.
5. Geben Sie weitere Nachrichten in das Producer-Fenster ein und beobachten Sie, wie sie im Consumer-Fenster angezeigt werden.

Nächster Schritt

Schritt 6: Löschen von Ressourcen

Schritt 6: Löschen von Ressourcen

In diesem Schritt löschen Sie die Ressourcen, die Sie in diesem Tutorial erstellt haben.

So löschen Sie den Cluster

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/home.
2. Wählen Sie in der Liste der Cluster den Cluster aus, den Sie für dieses Tutorial erstellt haben.
3. Wählen Sie für Aktionen die Option Cluster löschen.
4. Geben Sie `delete` in das Feld ein und wählen Sie dann Löschen.

So stoppen Sie den Client-Computer

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie in der Liste der Amazon-EC2-Instances den Client-Computer aus, den Sie für dieses Tutorial erstellt haben.
3. Wählen Sie Instance-Status und dann Instance beenden.
4. Wähen Sie Beenden.

So löschen Sie die IAM-Richtlinie und -Rolle

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Rollen.
3. Geben Sie in das Suchfeld den Namen der IAM-Rolle ein, die Sie für dieses Tutorial erstellt haben.
4. Wählen Sie die Rolle aus. Wählen Sie dann Rolle löschen und bestätigen Sie das Löschen.
5. Wählen Sie im Navigationsbereich Richtlinien.
6. Geben Sie in das Suchfeld den Namen der Richtlinie ein, die Sie für dieses Tutorial erstellt haben.
7. Wählen Sie die Richtlinie aus, um die zugehörige Übersichtsseite zu öffnen. Wählen Sie auf der Übersicht-Seite der Richtlinie die Option Richtlinie löschen.
8. Wählen Sie Löschen.

Konfiguration für Serverless-Cluster

Amazon MSK legt die Broker-Konfigurationseigenschaften für Serverless-Cluster fest. Sie können diese Konfigurationseigenschaft-Einstellungen des Brokers nicht ändern. Sie können jedoch die folgenden Themen-Konfigurationseigenschaften festlegen.

Konfigurationseigenschaft	Standard	Bearbeitbar	Maximal zulässiger Wert
cleanup.policy	Löschen	Ja, aber nur zum Zeitpunkt der Erstellung des Themas.	
compression.type	Produzent	Ja	
max.message.bytes	1048588	Ja	8 MiB
message.timestamp.difference.max.ms	long.max	Ja	
message.timestamp.type	CreateTime	Ja	
retention.bytes	250 GiB	Ja	250 GiB
retention.ms	7 Tage	Ja	Unbegrenzt

Sie können auch Apache-Kafka-Befehle verwenden, um Konfigurationseigenschaften auf Themenebene für neue und vorhandene Themen festzulegen oder zu ändern. Weitere Informationen zu Konfigurationseigenschaften auf Themenebene und Beispiele zum Festlegen dieser Eigenschaften finden Sie unter Konfigurationen auf Themenebene in der Apache-Kafka-Dokumentation.

Überwachen von Serverless-Clustern

Amazon MSK ist in Amazon integriert, CloudWatch sodass Sie Metriken für Ihren MSK-Serverless-Cluster sammeln, anzeigen und analysieren können. Die in der folgenden Tabelle aufgeführten

Metriken sind für alle Serverless-Cluster verfügbar. Da diese Metriken als einzelne Datenpunkte für jede Partition im Thema veröffentlicht werden, empfehlen wir, sie als SUM-Statistik zu betrachten, um eine Übersicht auf Themenebene zu erhalten.

Amazon MSK veröffentlicht `PerSec` Metriken mit einer CloudWatch Frequenz von einmal pro Minute. Das bedeutet, dass die SUM-Statistik für einen Zeitraum von einer Minute die Daten pro Sekunde für `PerSec`-Metriken genau wiedergibt. Verwenden Sie den folgenden CloudWatch mathematischen Ausdruck, um Daten pro Sekunde für einen Zeitraum von mehr als einer Minute zu sammeln:. `m1 * 60/PERIOD(m1)`

Auf der DEFAULT-Überwachungsebene verfügbare Metriken

Name	Wenn sichtbar	Dimensionen	Beschreibung
`BytesInPerSec`	Nachdem ein Produzent zu einem Thema geschrieben hat	Cluster-Name, Thema	Die Anzahl der Bytes, die pro Sekunde von Clients empfangen werden. Diese Metrik ist für jedes Thema verfügbar.
`BytesOutPerSec`	Nachdem eine Verbrauchergruppe von einem Thema konsumiert hat	Cluster-Name, Thema	Die Anzahl der Bytes, die pro Sekunde an Clients gesendet werden. Diese Metrik ist für jedes Thema verfügbar.
`FetchMessageConversionsPerSec`	Nachdem eine Verbrauchergruppe von einem Thema konsumiert hat	Cluster-Name, Thema	Die Anzahl der Abrufnachrichten-Konvertierungen pro Sekunde für den Broker.
`EstimatedMaxTimeLag`	Nachdem eine Verbrauchergruppe von einem Thema konsumiert hat	Cluster-Name, Verbrauchergruppe, Thema	Eine Zeitschätzung der MaxOffsetLag Metrik.
`MaxOffsetLag`	Nachdem eine Verbrauch	Cluster-Name,	Die maximale Offset-Verzögerung für alle Partitionen in einem Thema.

Name	Wenn sichtbar	Dimensionen	Beschreibung
	ergruppe von einem Thema konsumiert hat	Verbrauch ergruppe, Thema	
`MessagesInPerSec`	Nachdem ein Produzent zu einem Thema geschrieben hat	Cluster-Name, Thema	Die Anzahl der Nachrichten, die pro Sekunde für das Thema eingehen.
`ProduceMessageConversionsPerSec`	Nachdem ein Produzent zu einem Thema geschrieben hat	Cluster-Name, Thema	Die Anzahl der Produzenten-Nachrichtenkonvertierungen pro Sekunde für den Broker.
`SumOffsetLag`	Nachdem eine Verbrauch ergruppe von einem Thema konsumiert hat	Cluster-Name, Verbrauch ergruppe, Thema	Die aggregierte Offset-Verzögerung für alle Partitionen in einem Thema.

So zeigen Sie MSK.-Serverless-Metriken an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich unter Metriken Alle Metriken aus.
3. Suchen Sie in den Metriken nach dem Begriff **`kafka`**.
4. Wählen Sie AWS/Kafka / Cluster-Name, Thema oder AWS/Kafka / Cluster-Name, Verbrauchergruppe, Thema, um verschiedene Metriken anzuzeigen.

MSK Connect

Was ist MSK Connect?

MSK Connect ist ein Feature von Amazon MSK, die es Entwicklern erleichtert, Daten zu und von ihren Apache-Kafka-Clustern zu streamen. MSK Connect verwendet Kafka Connect 2.7.1, ein Open-Source-Framework für die Verbindung von Apache-Kafka-Clustern mit externen Systemen wie Datenbanken, Suchindizes und Dateisystemen. Mit MSK Connect können Sie vollständig verwaltete Konnektoren bereitstellen, die für Kafka Connect entwickelt wurden und Daten in beliebte Datenspeicher wie Amazon S3 und Amazon OpenSearch Service verschieben oder Daten aus diesen abrufen. Sie können Konnektoren einsetzen, die von Drittanbietern wie Debezium entwickelt wurden, um Änderungsprotokolle aus Datenbanken in einen Apache-Kafka-Cluster zu streamen, oder einen vorhandenen Konnektor ohne Codeänderungen bereitstellen. Konnektoren skalieren automatisch, um sich an Laständerungen anzupassen. Sie zahlen nur für die tatsächlich genutzten Ressourcen.

Verwenden Sie Quell-Konnektoren, um Daten aus externen Systemen in Ihre Themen zu importieren. Mit Sink-Konnektoren können Sie Daten aus Ihren Themen in externe Systeme exportieren.

MSK Connect unterstützt Konnektoren für jeden Apache-Kafka-Cluster mit Konnektivität zu einer Amazon VPC, unabhängig davon, ob es sich um einen MSK-Cluster oder einen unabhängig gehosteten Apache-Kafka-Cluster handelt.

MSK Connect überwacht kontinuierlich den Zustand und den Bereitstellungsstatus der Konnektoren, patcht und verwaltet die zugrunde liegende Hardware und skaliert die Konnektoren automatisch, um sie an Änderungen im Durchsatz anzupassen.

Die ersten Schritte mit MSK Connect finden Sie unter the section called “Erste Schritte”.

Informationen zu den AWS Ressourcen, die Sie mit MSK Connect erstellen können, finden Sie unter the section called “Konnektoren”the section called “Plug-ins”, undthe section called “Worker”.

Informationen zur MSK-Connect-API finden Sie in der Referenz zu Amazon MSK Connect API.

Erste Schritte mit MSK Connect

In diesem step-by-step Tutorial werden ein MSK-Cluster und ein Sink-Connector erstellt, der Daten vom Cluster an einen S3-Bucket sendet. AWS Management Console

Themen

Schritt 1: Die erforderlichen Ressourcen einrichten

In diesem Schritt erstellen Sie die folgenden Ressourcen, die Sie für dieses Erste-Schritte-Szenario benötigen:

- Ein S3-Bucket, der als Ziel dient und Daten vom Konnektor empfängt.
- Ein MSK-Cluster, an den Sie Daten senden werden. Der Konnektor liest dann die Daten aus diesem Cluster und sendet sie an den Ziel-S3-Bucket.
- Eine IAM-Rolle, die es dem Konnektor ermöglicht, in den S3-Ziel-Bucket zu schreiben.
- Ein Amazon-VPC-Endpunkt, der es ermöglicht, Daten von der Amazon-VPC, die den Cluster und den Konnektor enthält, an Amazon S3 zu senden.

So erstellen Sie den S3-Bucket

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/s3/.
2. Wählen Sie Bucket erstellen aus.
3. Geben Sie für den Namen des Buckets einen beschreibenden Namen ein, z. B. `mkc-tutorial-destination-bucket`.
4. Scrollen Sie nach unten und wählen Sie Bucket erstellen.
5. Wählen Sie in der Bucket-Liste den neu erstellten Bucket aus.
6. Wählen Sie Create folder.
7. Geben Sie `tutorial` für den Namen des Ordners ein, scrollen Sie dann nach unten und wählen Sie Ordner erstellen.

So erstellen Sie den Cluster

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/home?region=us-east-1#/home/.
2. Wählen Sie im linken Bereich unter MSK-Cluster die Option Cluster.
3. Wählen Sie Cluster erstellen.
4. Wählen Sie Benutzerdefiniert erstellen.
5. Geben Sie für Cluster-Name `mkc-tutorial-cluster` ein.
6. Wählen Sie unter Allgemeine Cluster-Eigenschaften Bereitgestellt als Cluster-Typ.
7. Wählen Sie unter Netzwerk eine Amazon VPC aus. Wählen Sie dann die Availability Zones und Subnetze aus, die Sie verwenden möchten. Merken Sie sich die IDs der Amazon VPC und der Subnetze, die Sie ausgewählt haben, da Sie sie später in diesem Tutorial benötigen.
8. Stellen Sie sicher, dass unter Zugriffssteuerungs-Methoden nur Nicht authentifizierter Zugriff ausgewählt ist.
9. Stellen Sie sicher, dass unter Verschlüsselung nur Klartext ausgewählt ist.
10. Fahren Sie mit dem Assistenten fort und wählen Sie dann Cluster erstellen. Dadurch gelangen Sie zur Detailseite für den Cluster. Suchen Sie auf dieser Seite unter Angewendete Sicherheitsgruppen nach der Sicherheitsgruppen-ID. Merken Sie sich diese ID, da Sie sie später in diesem Tutorial benötigen.

So erstellen Sie die IAM-Rolle, die in den Ziel-Bucket schreiben kann

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Bereich unter Zugriffsverwaltung die Option Rollen.
3. Wählen Sie Rolle erstellen aus.
4. Wählen Sie unter Oder Service auswählen, um Anwendungsfälle anzuzeigen die Option S3.
5. Scrollen Sie nach unten und wählen Sie unter Wählen Sie Ihren Anwendungsfall erneut S3.
6. Wählen Sie Next: Permissions aus.
7. Wählen Sie Richtlinie erstellen aus. Dadurch wird eine neue Registerkarte in Ihrem Browser geöffnet, auf der Sie die Richtlinie erstellen. Lassen Sie die ursprüngliche Registerkarte zur Rollenerstellung geöffnet, da wir später darauf zurückkommen werden.
8. Wählen Sie die Registerkarte JSON und ersetzen Sie dann den Text im Fenster durch die folgende Richtlinie.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets"
      ],
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation",
        "s3:DeleteObject"
      ],
      "Resource": "arn:aws:s3:::<my-tutorial-destination-bucket>"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:AbortMultipartUpload",
        "s3:ListMultipartUploadParts",
        "s3:ListBucketMultipartUploads"
      ],
      "Resource": "*"
    }
  ]
}
```

9. Wählen Sie Next: Tags (Weiter: Tags) aus.
10. Klicken Sie auf Weiter: Prüfen.
11. Geben Sie `mkc-tutorial-policy` für den Richtliniennamen ein und wählen Sie dann Richtlinie erstellen.
12. Zurück in der Browser-Registerkarte, in der Sie die Rolle erstellt haben, wählen Sie die Schaltfläche Aktualisieren.

13. Suchen Sie die `mkc-tutorial-policy` und wählen Sie sie aus, indem Sie die Schaltfläche links daneben wählen.
14. Wählen Sie Next: Tags (Weiter: Tags) aus.
15. Klicken Sie auf Weiter: Prüfen.
16. Geben Sie `mkc-tutorial-role` für den Rollennamen ein und löschen Sie den Text im Beschreibungsfeld.
17. Wählen Sie Rolle erstellen aus.

So erlauben Sie MSK Connect, die Rolle zu übernehmen

1. Wählen Sie in der IAM-Konsole im linken Bereich unter Zugriffsverwaltung die Option Rollen aus.
2. Suchen Sie die `mkc-tutorial-role` und wählen Sie sie aus.
3. Wählen Sie unter der Übersicht der Rolle die Registerkarte Vertrauensstellungen aus.
4. Wählen Sie Vertrauensstellung bearbeiten aus.
5. Ersetzen Sie die vorhandene Vertrauensrichtlinie durch den folgenden JSON-Code.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "kafkaconnect.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

6. Wählen Sie Update Trust Policy (Trust Policy aktualisieren).

So erstellen Sie einen Amazon-VPC-Endpunkt von der Cluster-VPC zu Amazon S3

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im linken Navigationsbereich Endpunkte aus.
3. Wählen Sie Endpunkt erstellen aus.

4. Wählen Sie unter Service-Name den Service com.amazonaws.us-east-1.s3 und den Gateway-Typ aus.
5. Wählen Sie die VPC des Clusters und dann das Feld links neben der Routing-Tabelle aus, die den Subnetzen des Clusters zugeordnet ist.
6. Wählen Sie Endpunkt erstellen aus.

Nächster Schritt

Schritt 2: Ein benutzerdefiniertes Plugin erstellen

Schritt 2: Ein benutzerdefiniertes Plugin erstellen

Ein Plugin enthält den Code, der die Logik des Konnektors definiert. In diesem Schritt erstellen Sie ein benutzerdefiniertes Plugin, das den Code für den Lenses Amazon S3 Sink Connector enthält. In einem späteren Schritt, wenn Sie den MSK-Konnektor erstellen, geben Sie an, dass sich sein Code in diesem benutzerdefinierten Plugin befindet. Sie können dasselbe Plugin verwenden, um mehrere MSK-Conectors mit unterschiedlichen Konfigurationen zu erstellen.

So erstellen Sie das benutzerdefinierte Plugin

1. Laden Sie den S3-Konnektor herunter.
2. Laden Sie die ZIP-Datei in einen S3-Bucket hoch, auf den Sie Zugriff haben. Informationen zum Hochladen von Dateien auf Amazon S3 finden Sie unter Hochladen von Objekten im Amazon-S3-Benutzerhandbuch.
3. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
4. Erweitern Sie im linken Bereich MSK Connect und wählen Sie dann Benutzerdefinierte Plugins.
5. Wählen Sie Benutzerdefiniertes Plugin erstellen.
6. Wählen Sie S3 durchsuchen.
7. Suchen Sie in der Liste der Buckets den Bucket, in den Sie die ZIP-Datei hochgeladen haben, und wählen Sie diesen Bucket aus.
8. Wählen Sie in der Liste der Objekte im Bucket das Optionsfeld links neben der ZIP-Datei aus und klicken Sie dann auf die Schaltfläche mit der Bezeichnung Auswählen.
9. Geben Sie `mkc-tutorial-plugin` für den Namen des benutzerdefinierten Plugins ein und wählen Sie dann Benutzerdefiniertes Plugin erstellen.

Es kann AWS einige Minuten dauern, bis die Erstellung des benutzerdefinierten Plugins abgeschlossen ist. Wenn der Erstellungsvorgang abgeschlossen ist, sehen Sie die folgende Meldung in einem Banner oben im Browserfenster.

```
Custom plugin mkc-tutorial-plugin was successfully created
The custom plugin was created. You can now create a connector using this custom plugin.
```

Nächster Schritt

Schritt 3: Client-Computer und Apache-Kafka-Thema erstellen

Schritt 3: Client-Computer und Apache-Kafka-Thema erstellen

In diesem Schritt erstellen Sie eine Amazon-EC2-Instance, die als Apache-Kafka-Client-Instance verwendet werden soll. Anschließend verwenden Sie diese Instance, um ein Thema im Cluster zu erstellen.

Erstellen eines Client-Computers

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie Launch Instances aus.
3. Geben Sie einen Namen für Ihren Client-Computer ein, z. B. **`mkc-tutorial-client`**
4. Lassen Sie Amazon Linux 2 AMI (HVM) – Kernel 5.10, SSD Volume Type als Amazon Machine Image (AMI)-Typ ausgewählt.
5. Wählen Sie den Instance-Typ t2.xlarge.
6. Wählen Sie unter Schlüsselpaar (Login) die Option Neues Schlüsselpaar erstellen. Geben Sie **`mkc-tutorial-key-pair`** für den Schlüsselpaar-Namen ein und wählen Sie dann Schlüsselpaar herunterladen. Alternativ können Sie ein vorhandenes Schlüsselpaar verwenden.
7. Wählen Sie Launch Instance (Instance starten) aus.
8. Klicken Sie auf View Instances (Instances anzeigen). Wählen Sie dann in der Spalte Sicherheitsgruppen die Sicherheitsgruppe, die Ihrer neuen Instance zugeordnet ist. Kopieren Sie die ID der Sicherheitsgruppe, und speichern Sie sie für später.

So erlauben Sie es dem neu erstellten Client, Daten an den Cluster zu senden

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.

2. Wählen Sie im linken Bereich unter SECURITY die Option Sicherheitsgruppen. Suchen Sie in der Spalte Sicherheitsgruppen-ID die Sicherheitsgruppe des Clusters. Sie haben die ID dieser Sicherheitsgruppe gespeichert, als Sie den Cluster in the section called “Schritt 1: Die erforderlichen Ressourcen einrichten” erstellt haben. Wählen Sie diese Sicherheitsgruppe aus, indem Sie das Feld links neben der Zeile auswählen. Stellen Sie sicher, dass keine anderen Sicherheitsgruppen gleichzeitig ausgewählt sind.
3. Wählen Sie im unteren Bereich der Seite die Registerkarte Regeln für eingehenden Datenverkehr.
4. Wählen Sie Edit inbound rules (Regeln für eingehenden Datenverkehr bearbeiten) aus.
5. Wählen Sie unten links auf dem Bildschirm Regel hinzufügen.
6. Wählen Sie in der neuen Regel All traffic (Gesamter Datenverkehr) in der Spalte Type (Typ). Geben Sie im Feld rechts neben der Spalte Quelle die ID der Sicherheitsgruppe des Client-Computers ein. Dies ist die Sicherheitsgruppen-ID, die Sie gespeichert haben, nachdem Sie den Client-Computer erstellt haben.
7. Wählen Sie Save rules (Regeln speichern) aus. Ihr MSK-Cluster akzeptiert jetzt den gesamten Datenverkehr von dem Client, den Sie im vorherigen Verfahren erstellt haben.

Erstellen Sie ein Thema wie folgt

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie `mkc-tutorial-client` in der Instance-Tabelle.
3. Wählen Sie oben auf dem Bildschirm Verbinden aus und folgen Sie dann den Anweisungen, um eine Verbindung mit der Instance herzustellen.
4. Installieren Sie Java auf der Client-Instance, indem Sie den folgenden Befehl ausführen:

```
sudo yum install java-1.8.0
```

5. Führen Sie den folgenden Befehl aus, um Apache Kafka herunterzuladen.

```
wget https://archive.apache.org/dist/kafka/2.2.1/kafka_2.12-2.2.1.tgz
```

> **Note**
>
> Wenn Sie eine andere als die in diesem Befehl verwendete Spiegelsite verwenden möchten, können Sie eine andere auf der Apache-Website auswählen.

6. Führen Sie den folgenden Befehl in dem Verzeichnis aus, in das Sie im vorherigen Schritt die TAR-Datei heruntergeladen haben.

```
tar -xzf kafka_2.12-2.2.1.tgz
```

7. Wechseln Sie zum Verzeichnis kafka_2.12-2.2.1.
8. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/home?region=us-east-1#/home/.
9. Wählen Sie im linken Bereich Cluster und dann den Namen `mkc-tutorial-cluster`.
10. Wählen Sie Client-Informationen anzeigen aus.
11. Kopieren Sie die Klartext-Verbindungszeichenfolge.
12. Wählen Sie Erledigt aus.
13. Führen Sie den folgenden Befehl auf der Client-Instanz (`mkc-tutorial-client`) aus und *`bootstrapServerString`*ersetzen Sie ihn durch den Wert, den Sie gespeichert haben, als Sie sich die Client-Informationen des Clusters angesehen haben.

```
<path-to-your-kafka-installation>/bin/kafka-topics.sh --create --bootstrap-
server bootstrapServerString --replication-factor 2 --partitions 1 --topic mkc-
tutorial-topic
```

Wenn der Befehl erfolgreich ist, wird die folgende Meldung angezeigt: `Created topic mkc-tutorial-topic.`

Nächster Schritt

Schritt 4: Konnektor erstellen

Schritt 4: Konnektor erstellen

So erstellen Sie den Konnektor

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon MSK-Konsole unter https://console.aws.amazon.com/msk/home?region=us-east-1#/home/.
2. Wählen Sie im linken Bereich unter MSK Connect die Option Konnektoren.
3. Wählen Sie Konnektor erstellen.

4. Wählen Sie in der Liste der Plugins die Option `mkc-tutorial-plugin` und anschließend Weiter.
5. Geben Sie als Namen des Konnektors `mkc-tutorial-connector` ein.
6. Wählen Sie in der Liste der Cluster `mkc-tutorial-cluster`.
7. Kopieren Sie die folgende Konfiguration und fügen Sie sie in das Feld für die Konnektor-Konfiguration ein.

```
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=<my-tutorial-destination-bucket>
topics.dir=tutorial
```

8. Wählen Sie unter Zugriffsberechtigungen die Option `mkc-tutorial-role`.
9. Wählen Sie Weiter aus. Wählen Sie auf der Seite Sicherheit erneut Weiter.
10. Wählen Sie auf der Seite Protokolle Weiter.
11. Wählen Sie unter Überprüfen und erstellen die Option Konnektor erstellen.

Nächster Schritt

Schritt 5: Daten senden

Schritt 5: Daten senden

In diesem Schritt senden Sie Daten an das Apache-Kafka-Thema, das Sie zuvor erstellt haben, und suchen dann im Ziel-S3-Bucket nach denselben Daten.

So senden Sie Daten an den MSK-Cluster

1. Wenn Sie sich noch im `bin`-Ordner der Apache-Kafka-Installation auf der Client-Instance befinden, erstellen Sie eine Textdatei namens `client.properties` mit dem folgenden Inhalt.

```
security.protocol=PLAINTEXT
```

2. Führen Sie den folgenden Befehl aus, um einen Konsolenproduzenten zu erstellen. *BootstrapBrokerString*Ersetzen Sie es durch den Wert, den Sie bei der Ausführung des vorherigen Befehls erhalten haben.

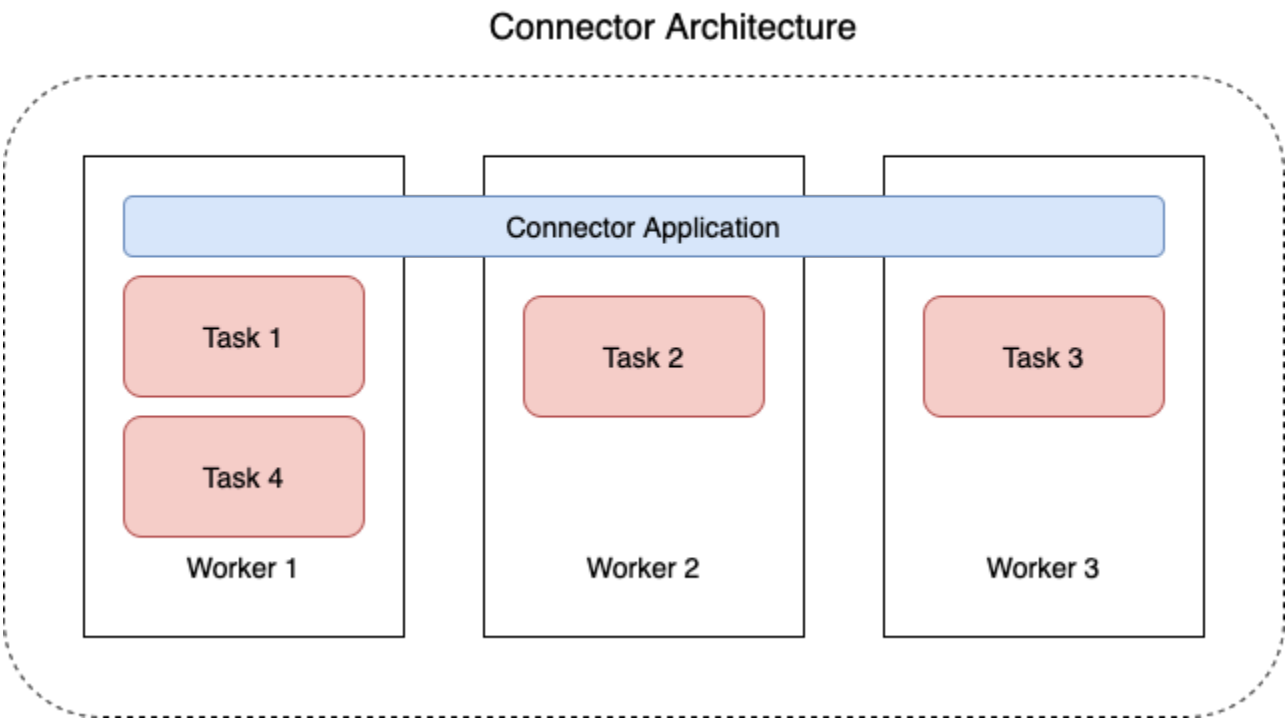
```
<path-to-your-kafka-installation>/bin/kafka-console-producer.sh --broker-
list BootstrapBrokerString --producer.config client.properties --topic mkc-
tutorial-topic
```

3. Geben Sie eine beliebige Nachricht ein, und drücken Sie Enter (Eingabetaste). Wiederholen Sie diesen Schritt zwei- oder dreimal. Jedes Mal, wenn Sie eine Zeile eingeben und Enter (Eingabetaste) drücken, wird diese Zeile als separate Nachricht an Ihren Apache Kafka-Cluster gesendet.
4. Suchen Sie im Amazon-S3-Ziel-Bucket nach den Nachrichten, die Sie im vorherigen Schritt gesendet haben.

Konnektoren

Ein Konnektor integriert externe Systeme und Amazon-Services mit Apache Kafka, indem er kontinuierlich Streaming-Daten aus einer Datenquelle in Ihren Apache-Kafka-Cluster kopiert oder kontinuierlich Daten aus Ihrem Cluster in einen Daten-Sink kopiert. Ein Konnektor kann auch einfache Logik wie Transformation, Formatkonvertierung oder Filterung von Daten ausführen, bevor die Daten an ein Ziel gesendet werden. Quell-Konnektoren rufen Daten aus einer Datenquelle ab und übertragen diese Daten in den Cluster, während Sink-Konnektoren Daten aus dem Cluster abrufen und diese Daten in einen Daten-Sink übertragen.

Das folgende Diagramm illustriert die Architektur eines Konnektors. Ein Worker ist ein virtueller Java-Maschine (JVM)-Prozess, der die Konnektor-Logik betreibt. Jeder Worker erstellt eine Reihe von Aufgaben, die in parallelen Threads ausgeführt werden und das Kopieren der Daten übernehmen. Aufgaben speichern keinen Status und können daher jederzeit gestartet, gestoppt oder neu gestartet werden, um eine stabile und skalierbare Datenpipeline bereitzustellen.



Kapazität des Konnektors

Die Gesamtkapazität eines Konnektors hängt von der Anzahl der Worker des Konnektors sowie von der Anzahl der MSK Connect Units (MCUs) pro Worker ab. Jede MCU steht für 1 vCPU Rechenleistung und 4 GiB Arbeitsspeicher. Der MCU-Speicher bezieht sich auf den Gesamtspeicher einer Worker-Instance und nicht auf den verwendeten Heap-Speicher.

MSK Connect-Mitarbeiter verwenden IP-Adressen in den vom Kunden bereitgestellten Subnetzen. Jeder Mitarbeiter verwendet eine IP-Adresse aus einem der vom Kunden bereitgestellten Subnetze. Sie sollten sicherstellen, dass in den Subnetzen, die für eine CreateConnector Anfrage bereitgestellt werden, genügend IP-Adressen verfügbar sind, um deren angegebene Kapazität zu berücksichtigen, insbesondere bei der automatischen Skalierung von Connectoren, bei denen die Anzahl der Worker schwanken kann.

Um einen Konnektor zu erstellen, müssen Sie zwischen einem der folgenden beiden Kapazitätsmodi wählen.

- Bereitgestellt – Wählen Sie diesen Modus, wenn Sie die Kapazitätsanforderungen für Ihren Konnektor kennen. Sie geben zwei Werte an:
  - Die Anzahl der Worker.
  - Die Anzahl der MCUs pro Worker.

- Automatisch skaliert – Wählen Sie diesen Modus, wenn die Kapazitätsanforderungen für Ihren Konnektor variabel sind oder wenn Sie sie nicht im Voraus kennen. Bei Verwendung des automatisch skalierten Kapazitätsmodus überschreibt Amazon MSK Connect die `tasks.max`-Eigenschaft des Konnektors mit einem Wert, der proportional zur Anzahl der Worker, die im Konnektor laufen, und zur Anzahl der MCUs pro Worker ist.

  Sie geben drei Wertesätze an:

  - Die minimale und maximale Anzahl von Workers.
  - Die Prozentsätze des Ab- und Aufskalierens der CPU-Auslastung, die durch die Metrik `CpuUtilization` bestimmt werden. Wenn die `CpuUtilization`-Metrik für den Konnektor den Aufskalier-Prozentsatz überschreitet, erhöht MSK Connect die Anzahl der Worker, die im Konnektor laufen. Wenn die `CpuUtilization`-Metrik unter den Abskalierungsprozentsatz fällt, verringert MSK Connect die Anzahl der Worker. Die Anzahl der Worker bleibt immer innerhalb der Mindest- und Höchstwerte, die Sie bei der Erstellung des Konnektors angeben.
  - Die Anzahl der MCUs pro Worker.

Weitere Informationen zu Worker finden Sie unter the section called “Worker”. Weitere Informationen zu MSK-Connect-Metriken finden Sie unter the section called “Überwachen”.

Erstellen eines Konnektors

Erstellen eines Connectors mit dem AWS Management Console

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
2. Wählen Sie im linken Bereich unter MSK Connect die Option Konnektoren.
3. Wählen Sie Konnektor erstellen).
4. Sie können wählen, ob Sie ein vorhandenes benutzerdefiniertes Plugin verwenden möchten, um den Konnektor zu erstellen, oder ob Sie zuerst ein neues benutzerdefiniertes Plugin erstellen möchten. Informationen zu benutzerdefinierten Plugins und deren Erstellung finden Sie unter the section called “Plug-ins”. Gehen wir bei diesem Verfahren davon aus, dass Sie über ein benutzerdefiniertes Plugin verfügen, das Sie verwenden möchten. Suchen Sie in der Liste der benutzerdefinierten Plugins nach dem Plugin, das Sie verwenden möchten, wählen Sie das Kästchen links davon aus und dann Weiter.
5. Geben Sie einen Namen und optional eine Beschreibung ein.
6. Wählen Sie den Cluster, zu dem Sie eine Verbindung herstellen möchten.

7. Geben Sie die Konnektor-Konfiguration an. Die Konfigurationsparameter, die Sie angeben müssen, hängen vom Typ des Konnektors ab, den Sie erstellen möchten. Einige Parameter sind jedoch allen Konnektoren gemeinsam, z. B. die Parameter `connector.class` und `tasks.max`. Im Folgenden finden Sie eine Beispielkonfiguration für den Confluent Amazon S3 Sink Connector.

```
connector.class=io.confluent.connect.s3.S3SinkConnector
tasks.max=2
topics=my-example-topic
s3.region=us-east-1
s3.bucket.name=my-destination-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. Als Nächstes konfigurieren Sie die Kapazität Ihres Konnektors. Sie können zwischen zwei Kapazitätsmodi wählen: bereitgestellt und automatisch skaliert. Weitere Informationen zu diesen beiden Optionen finden Sie unter the section called “Capacity (Kapazität)”.
9. Wählen Sie entweder die Standard-Worker-Konfiguration oder eine benutzerdefinierte Worker-Konfiguration. Weitere Informationen zum Erstellen von benutzerdefinierten Worker-Konfigurationen finden Sie unter the section called “Worker”.
10. Geben Sie als nächstes die Service-Ausführungsrolle an. Dies muss eine IAM-Rolle sein, die MSK Connect übernehmen kann und die dem Connector alle Berechtigungen gewährt, die er für den Zugriff auf die erforderlichen AWS Ressourcen benötigt. Diese Berechtigungen hängen von der Logik des Konnektors ab. Weitere Informationen zum Erstellen dieser Rolle finden Sie unter the section called “Service-Ausführungsrolle”.
11. Wählen Sie Weiter, überprüfen Sie die Sicherheitsinformationen und wählen Sie dann erneut Weiter.
12. Geben Sie die gewünschten Protokollierungs-Optionen an und wählen Sie dann Weiter. Weitere Informationen zur Protokollierung finden Sie unter the section called “Protokollierung”.
13. Wählen Sie Konnektor erstellen.

Informationen zur Verwendung der MSK Connect-API zum Erstellen eines Connectors finden Sie unter CreateConnector.

Plug-ins

Ein Plugin ist eine AWS Ressource, die den Code enthält, der Ihre Konnektorlogik definiert. Sie laden eine JAR-Datei (oder eine ZIP-Datei, die eine oder mehrere JAR-Dateien enthält) in einen S3-Bucket hoch und geben den Speicherort des Buckets an, wenn Sie das Plugin erstellen. Wenn Sie einen Konnektor erstellen, geben Sie das Plugin an, das MSK Connect dafür verwenden soll. Das Verhältnis von Plugins zu Konnektoren ist one-to-many: Sie können einen oder mehrere Konnektoren aus demselben Plugin erstellen.

Informationen zur Entwicklung des Codes für einen Konnektor finden Sie im Konnektor-Entwicklerleitfaden in der Apache-Kafka-Dokumentation.

Erstellen eines benutzerdefinierten Plugins mit dem AWS Management Console

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
2. Wählen Sie im linken Bereich unter MSK Connect die Option Benutzerdefinierte Plugins.
3. Wählen Sie Benutzerdefiniertes Plugin erstellen.
4. Wählen Sie S3 durchsuchen.
5. Wählen Sie in der Liste der S3-Buckets den Bucket aus, der die JAR- oder ZIP-Datei für das Plugin enthält.
6. Aktivieren Sie in der Objektliste das Kontrollkästchen links neben der JAR- oder ZIP-Datei für das Plug-in und wählen Sie dann Auswählen.
7. Wählen Sie Benutzerdefiniertes Plugin erstellen.

Informationen zur Verwendung der MSK Connect-API zum Erstellen eines benutzerdefinierten Plug-ins finden Sie unter CreateCustomPlugin.

Worker

Ein Worker ist ein virtueller Java-Maschine (JVM)-Prozess, der die Konnektor-Logik betreibt. Jeder Worker erstellt eine Reihe von Aufgaben, die in parallelen Threads ausgeführt werden und das Kopieren der Daten übernehmen. Aufgaben speichern keinen Status und können daher jederzeit gestartet, gestoppt oder neu gestartet werden, um eine stabile und skalierbare

Datenpipeline bereitzustellen. Änderungen an der Anzahl der Worker, unabhängig davon, ob sie auf ein Skalierungsereignis oder auf unerwartete Ausfälle zurückzuführen sind, werden von den verbleibenden Workern automatisch erkannt. Sie koordinieren, um die Aufgaben auf die Gruppe der verbleibenden Worker neu auszurichten. Connect-Worker nutzen die Verbrauchergruppen von Apache Kafka, um sich zu koordinieren und das Gleichgewicht wiederherzustellen.

Wenn die Kapazitätsanforderungen Ihres Konnektors variabel oder schwer abzuschätzen sind, können Sie MSK Connect die Anzahl der Worker nach Bedarf zwischen einer von Ihnen angegebenen Untergrenze und einer Obergrenze skalieren lassen. Sie können auch die genaue Anzahl von Workern angeben, die die Konnektor-Logik betreiben sollen. Weitere Informationen finden Sie unter the section called “Capacity (Kapazität)”.

MSK Connect-Mitarbeiter verbrauchen IP-Adressen

MSK Connect-Mitarbeiter verwenden IP-Adressen in den vom Kunden bereitgestellten Subnetzen. Jeder Mitarbeiter verwendet eine IP-Adresse aus einem der vom Kunden bereitgestellten Subnetze. Sie sollten sicherstellen, dass in den Subnetzen, die für eine CreateConnector Anfrage bereitgestellt werden, genügend IP-Adressen verfügbar sind, um deren angegebene Kapazität zu berücksichtigen, insbesondere bei der automatischen Skalierung von Connectoren, bei denen die Anzahl der Worker schwanken kann.

Themen

Standard-Worker-Konfiguration

MSK Connect bietet die folgende Standard-Worker-Konfiguration:

```
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.kafka.connect.storage.StringConverter
```

Unterstützte Worker-Konfigurationseigenschaften

MSK Connect bietet eine Standard-Worker-Konfiguration. Sie haben auch die Möglichkeit, eine benutzerdefinierte Worker-Konfiguration zur Verwendung mit Ihren Konnektoren zu erstellen. Die

folgende Liste enthält Informationen zu den Worker-Konfigurationseigenschaften, die Amazon MSK Connect unterstützt oder nicht unterstützt.

- Es werden die Eigenschaften `key.converter` und `value.converter` benötigt.
- MSK Connect unterstützt die folgenden `producer.`-Konfigurationseigenschaften.

```
producer.acks
producer.batch.size
producer.buffer.memory
producer.compression.type
producer.enable.idempotence
producer.key.serializer
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 unterstützt die folgenden `consumer.`-Konfigurationseigenschaften.

```
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.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
```

- Alle anderen Konfigurationseigenschaften, die nicht mit den Präfixen `producer.` oder `consumer.` beginnen, werden unterstützt, mit Ausnahme der folgenden Eigenschaften.

```
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
```

Weitere Informationen zu Worker-Konfigurationen und was sie bedeuten, finden Sie unter Kafka Connect Configs in der Apache-Kafka-Dokumentation.

Erstellen einer benutzerdefinierten Worker-Konfiguration

Erstellen einer benutzerdefinierten Worker-Konfiguration mit dem AWS Management Console

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
2. Wählen Sie im linken Bereich unter MSK Connect die Option Worker-Konfigurationen.
3. Wählen Sie Worker-Konfiguration erstellen.
4. Geben Sie einen Namen und eine optionale Beschreibung ein und fügen Sie dann die Eigenschaften und Werte hinzu, auf die Sie diese festlegen möchten.
5. Wählen Sie Worker-Konfiguration erstellen.

Informationen zur Verwendung der MSK Connect-API zum Erstellen einer Worker-Konfiguration finden Sie unter CreateWorkerConfiguration.

Verwaltung von Quell-Konnektor-Offsets mit `offset.storage.topic`

In diesem Abschnitt finden Sie Informationen zur Verwaltung von Quell-Konnektor-Offsets mithilfe des Offset-Speicherthemas. Das Offset-Speicherthema ist ein internes Thema, das Kafka Connect verwendet, um Offsets der Konnektor- und Aufgaben-Konfiguration zu speichern.

Es wird das standardmäßige Offset-Speicherthema verwendet

Standardmäßig generiert Amazon MSK Connect für jeden Konnektor, den Sie erstellen, ein neues Offset-Speicherthema in Ihrem Kafka-Cluster. MSK erstellt den Standard-Themennamen unter Verwendung von Teilen des Konnektor-ARN. z. B. `__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-4abc-8be8-c657f7e4ff32-2`.

Festlegen Ihres eigenen Offset-Speicherthemas

Um die Offset-Kontinuität zwischen den Quell-Konnektoren zu gewährleisten, können Sie anstelle des Standardthemas ein Offset-Speicherthema Ihrer Wahl verwenden. Wenn Sie ein Offset-Speicherthema angeben, können Sie Aufgaben wie das Erstellen eines Quell-Konnektors erledigen, der den Lesevorgang vom letzten Offset eines vorherigen Konnektors aus wieder aufnimmt.

Um ein Offset-Speicherthema anzugeben, geben Sie einen Wert für die Eigenschaft `offset.storage.topic` in Ihrer Worker-Konfiguration ein, bevor Sie einen Konnektor erstellen. Wenn Sie das Offset-Speicherthema wiederverwenden möchten, um Offsets von einem zuvor

erstellten Konnektor zu verwenden, müssen Sie dem neuen Konnektor denselben Namen wie dem alten Konnektor geben. Wenn Sie ein benutzerdefiniertes Offset-Speicherthema erstellen, müssen Sie `cleanup.policy` in Ihrer Themenkonfiguration auf `compact` einstellen.

> **Note**
>
> Wenn Sie beim Erstellen eines Sink-Konnektors ein Offset-Speicherthema angeben, erstellt MSK Connect das Thema, sofern es noch nicht vorhanden ist. Das Thema wird jedoch nicht zum Speichern von Konnektor-Offsets verwendet.
> Sink-Konnektor-Offsets werden stattdessen mithilfe des Kafka-Verbrauchergruppen-Protokolls verwaltet. Jeder Sink-Konnektor erstellt eine Gruppe mit dem Namen `connect-{CONNECTOR_NAME}`. Solange die Verbrauchergruppe existiert, werden alle aufeinanderfolgenden Sink-Konnektoren, die Sie mit demselben Wert für `CONNECTOR_NAME` erstellen, ab dem letzten festgeschriebenen Offset fortgesetzt.

Example : Angabe eines Offset-Speicherthemas, um einen Quell-Konnektor mit einer aktualisierten Konfiguration neu zu erstellen

Angenommen, Sie haben einen Change Data Capture (CDC)-Konnektor und möchten die Konnektor-Konfiguration ändern, ohne Ihren Platz im CDC-Stream zu verlieren. Sie können die bestehende Konnektor-Konfiguration nicht aktualisieren, aber Sie können den Konnektor löschen und einen neuen mit demselben Namen erstellen. Um dem neuen Konnektor mitzuteilen, wo er mit dem Lesen im CDC-Stream beginnen soll, können Sie das Offset-Speicherthema des alten Konnektors in Ihrer Worker-Konfiguration angeben. In den folgenden Schritten wird gezeigt, wie Sie diese Aufgabe erfüllen.

1. Führen Sie auf Ihrem Client-Computer den folgenden Befehl aus, um den Namen des Offset-Speicherthemas Ihres Konnektors zu ermitteln. Ersetzen Sie *<bootstrapBrokerString>* durch den Bootstrap-Broker-String Ihres Clusters. Anleitungen zum Abrufen des Bootstrap-Broker-Strings finden Sie unter Abrufen der Bootstrap-Broker für einen Amazon-MSK-Cluster.

   ```
   <path-to-your-kafka-installation>/bin/kafka-topics.sh --list --bootstrap-
   server <bootstrapBrokerString>
   ```

   Die folgende Ausgabe zeigt eine Liste aller Cluster-Themen, einschließlich aller standardmäßigen internen Konnektor-Themen. In diesem Beispiel verwendet der vorhandene CDC-Konnektor das von MSK Connect erstellte Standard-Offset-Speicherthema. Aus diesem

Grund wird das Offset-Speicherthema `__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-4abc-8be8-c657f7e4ff32-2` genannt.

```
__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. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msks/.
3. Wählen Sie Ihren Konnektor aus der Konnektoren-Liste aus. Kopieren und speichern Sie den Inhalt des Felds Konnektor-Konfiguration, sodass Sie ihn ändern und zum Erstellen des neuen Konnektors verwenden können.
4. Wählen Sie Löschen, um den Konnektor zu löschen. Geben Sie dann den Konnektor-Namen in das Texteingabefeld ein, um den Löschvorgang zu bestätigen.
5. Erstellen Sie eine benutzerdefinierte Worker-Konfiguration mit Werten, die zu Ihrem Szenario passen. Anweisungen finden Sie unter Erstellen einer benutzerdefinierten Worker-Konfiguration.

   In Ihrer Worker-Konfiguration müssen Sie den Namen des Offset-Speicherthemas, das Sie zuvor abgerufen haben, als Wert für `offset.storage.topic` angeben, wie in der folgenden Konfiguration.

   ```
   config.providers.secretManager.param.aws.region=us-east-1
   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**

   Sie müssen dem neuen Konnektor denselben Namen wie dem alten Konnektor geben.

Erstellen Sie einen neuen Konnektor mit der Worker-Konfiguration, die Sie im vorherigen Schritt eingerichtet haben. Anweisungen finden Sie unter Erstellen eines Konnektors.

Überlegungen

Beachten Sie Folgendes, wenn Sie die Quell-Konnektor-Offsets verwalten.

- Um ein Offset-Speicherthema anzugeben, geben Sie den Namen des Kafka-Themas, in dem Konnektor-Offsets gespeichert werden, als Wert für `offset.storage.topic` in Ihrer Worker-Konfiguration an.
- Seien Sie vorsichtig, wenn Sie Änderungen an einer Konnektor-Konfiguration vornehmen. Das Ändern von Konfigurationswerten kann zu unbeabsichtigtem Verhalten des Konnektors führen, wenn ein Quell-Konnektor Werte aus der Konfiguration für wichtige Offset-Datensätze verwendet. Wir empfehlen Ihnen, in der Dokumentation Ihres Plugins nach Anleitungen zu suchen.
- Anpassen der Standardanzahl von Partitionen – Sie können nicht nur die Worker-Konfiguration durch Hinzufügen von `offset.storage.topic` anpassen, sondern auch die Anzahl der Partitionen für die Offset- und Status-Speicherthemen anpassen. Die Standardpartitionen für interne Themen lauten wie folgt.
  - `config.storage.topic`: 1, nicht konfigurierbar, muss ein Thema mit einer einzigen Partition sein
  - `offset.storage.topic`: 25, konfigurierbar durch Bereitstellung von `offset.storage.partitions`
  - `status.storage.topic`: 5, konfigurierbar durch Bereitstellung von `status.storage.partitions`
- Manuelles Löschen von Themen – Amazon MSK Connect erstellt bei jeder Bereitstellung von Konnektoren neue interne Kafka-Connect-Themen (der Themenname beginnt mit `__amazon_msk_connect`). Alte Themen, die an gelöschte Konnektoren angehängt sind, werden nicht automatisch entfernt, da interne Themen, wie z. B. `offset.storage.topic`, zwischen Konnektoren wiederverwendet werden können. Sie können jedoch nicht verwendete interne Themen, die von MSK Connect erstellt wurden, manuell löschen. Die internen Themen sind nach dem Format `__amazon_msk_connect_<offsets|status|configs>_`*`connector_name_connector_id`* benannt.

Der reguläre Ausdruck `__amazon_msk_connect_<offsets|status|configs>_connector_name_connector_id` kann verwendet werden, um die internen Themen zu löschen. Sie sollten kein internes Thema löschen, das derzeit von einem laufenden Konnektor verwendet wird.

- Den selben Namen für die von MSK Connect erstellten internen Themen – Wenn Sie das Offset-Speicherthema wiederverwenden möchten, um Offsets von einem zuvor erstellten Konnektor zu verwenden, müssen Sie dem neuen Konnektor denselben Namen wie dem alten Konnektor geben. Die `offset.storage.topic` Eigenschaft kann mithilfe der Worker-Konfiguration festgelegt werden, um dem `offset.storage.topic` denselben Namen zuzuweisen, und zwischen verschiedenen Konnektoren wiederverwendet werden. Diese Konfiguration wird unter Konnektor-Offsets verwalten beschrieben. MSK Connect erlaubt nicht, dass verschiedene Konnektoren `config.storage.topic` und `status.storage.topic` gemeinsam nutzen. Diese Themen werden jedes Mal erstellt, wenn Sie einen neuen Konnektor in MSKC erstellen. Sie werden automatisch nach dem Format `__amazon_msk_connect_<status|configs>_connector_name_connector_id` benannt und unterscheiden sich daher bei den verschiedenen Konnektoren, die Sie erstellen.

Externalisierung vertraulicher Informationen mithilfe von Konfigurationsanbietern

Dieses Beispiel zeigt, wie vertrauliche Informationen für Amazon MSK Connect mithilfe eines Open-Source-Konfigurationsanbieters externalisiert werden. Mit einem Konfigurationsanbieter können Sie Variablen anstelle von Klartext in einer Konnektor- oder Worker-Konfiguration angeben, und Worker, die im Konnektor ausgeführt werden, lösen diese Variablen zur Laufzeit auf. Dadurch wird verhindert, dass Anmeldeinformationen und andere Secrets im Klartext gespeichert werden. Der Konfigurationsanbieter im Beispiel unterstützt das Abrufen von Konfigurationsparametern von AWS Secrets Manager, Amazon S3 und Systems Manager (SSM). In Schritt 2 erfahren Sie, wie Sie das Speichern und Abrufen vertraulicher Informationen für den Service einrichten, den Sie konfigurieren möchten.

Themen

- Schritt 1: Ein benutzerdefiniertes Plugin erstellen und auf S3 hochladen
- Schritt 2: Parameter und Berechtigungen für verschiedene Anbieter konfigurieren

- Schritt 3: Erstellen Sie eine benutzerdefinierte Worker-Konfiguration mit Informationen zu Ihrem Konfigurationsanbieter
- Schritt 4: Den Konnektor erstellen
- Überlegungen

Schritt 1: Ein benutzerdefiniertes Plugin erstellen und auf S3 hochladen

Um ein benutzerdefiniertes Plugin zu erstellen, erstellen Sie eine ZIP-Datei, die den Connector enthält, und führen Sie die msk-config-provider folgenden Befehle auf Ihrem lokalen Computer aus.

So erstellen Sie ein benutzerdefiniertes Plugin mit einem Terminalfenster und Debezium als Konnektor

Verwenden Sie die AWS CLI, um Befehle als Superuser mit Anmeldeinformationen auszuführen, mit denen Sie auf Ihren AWS S3-Bucket zugreifen können. Informationen zur Installation und Einrichtung der AWS CLI finden Sie unter Erste Schritte mit der AWS CLI im AWS Command Line Interface Benutzerhandbuch. Informationen zur Verwendung der AWS CLI mit Amazon S3 finden Sie unter Verwenden von Amazon S3 mit der AWS CLI im AWS Command Line Interface Benutzerhandbuch.

1. Erstellen Sie in einem Terminal-Fenster mit dem folgenden Befehl einen Ordner mit dem Namen `custom-plugin` in Ihrem Workspace.

   ```
   mkdir custom-plugin && cd custom-plugin
   ```

2. Laden Sie die neueste stabile Version des MySQL-Konnektor-Plugins mit dem folgenden Befehl von der Debezium-Website herunter.

   ```
   wget https://repo1.maven.org/maven2/io/debezium/debezium-connectormysql/
   2.2.0.Final/debezium-connector-mysql-2.2.0.Final-plugin.tar.gz
   ```

   Extrahieren Sie die heruntergeladene GZIP-Datei mit dem folgenden Befehl in den Ordner `custom-plugin`.

   ```
   tar xzf debezium-connector-mysql-2.2.0.Final-plugin.tar.gz
   ```

3. Laden Sie die ZIP-Datei des MSK-Konfigurationsanbieters mit dem folgenden Befehl herunter.

```
wget https://github.com/aws-samples/msk-config-providers/releases/download/r0.1.0/
msk-config-providers-0.1.0-with-dependencies.zip
```

Extrahieren Sie die heruntergeladene GZIP-Datei mit dem folgenden Befehl in den Ordner `custom-plugin`.

```
unzip msk-config-providers-0.1.0-with-dependencies.zip
```

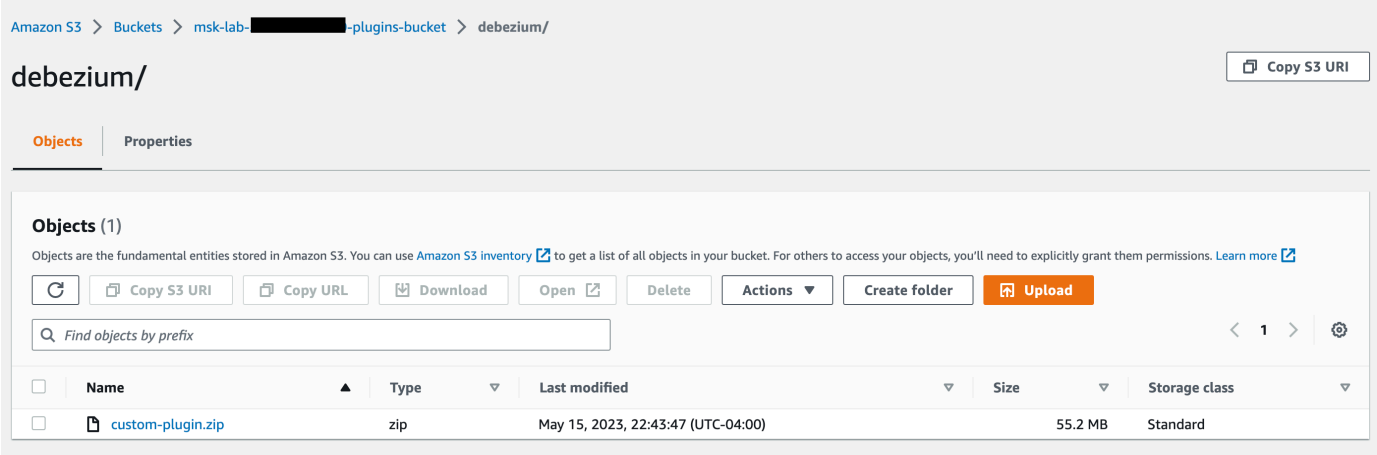
4. Komprimieren Sie den Inhalt des MSK-Konfigurationsanbieters aus dem obigen Schritt und den benutzerdefinierten Konnektor in einer einzigen Datei mit dem Namen `custom-plugin.zip`.

```
zip -r ../custom-plugin.zip *
```

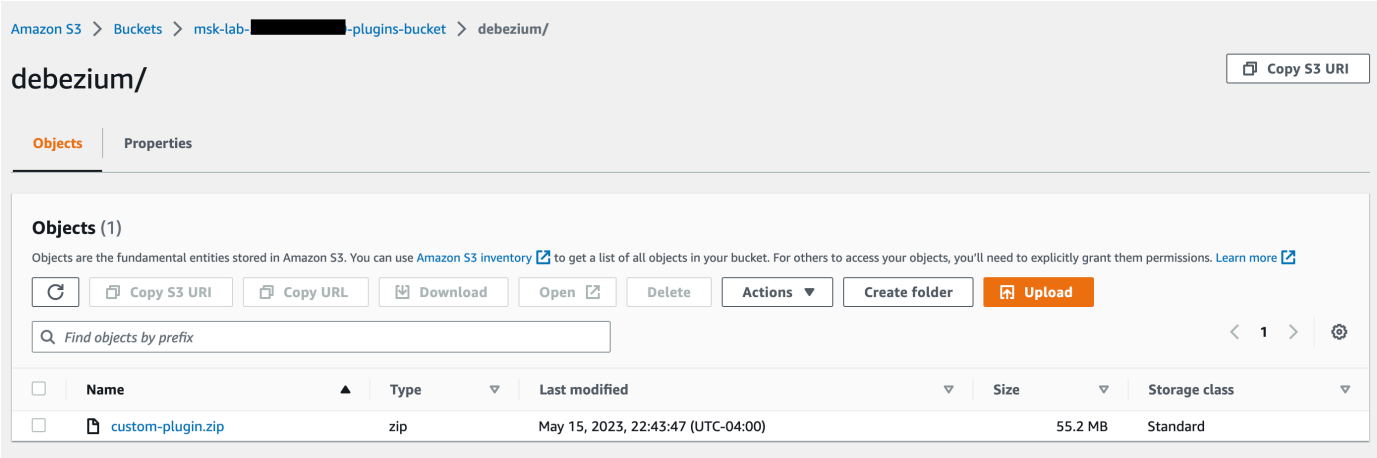
5. Laden Sie die Datei auf S3 hoch, damit sie später referenziert werden kann.

```
aws s3 cp ../custom-plugin.zip s3:<S3_URI_BUCKET_LOCATION>
```

6. Wählen Sie auf der Amazon-MSK-Konsole im Abschnitt MSK Connect die Option Benutzerdefiniertes Plugin und dann Benutzerdefiniertes Plugin erstellen. Durchsuchen Sie den S3-Bucket unter s3:<***S3_URI_BUCKET_LOCATION***>, um die benutzerdefinierte Plugin-ZIP-Datei auszuwählen, die Sie gerade hochgeladen haben.



7. Geben Sie für den Namen des Plugins **`debezium-custom-plugin`** ein. Geben Sie optional eine Beschreibung ein und wählen Sie Benutzerdefiniertes Plugin erstellen.



Schritt 2: Parameter und Berechtigungen für verschiedene Anbieter konfigurieren

Sie können Parameterwerte in diesen drei Services konfigurieren:

- Secrets Manager
- Systems Manager Parameter Store
- S3 - Simple Storage Service

Wählen Sie eine der folgenden Registerkarten aus, um Anweisungen zur Einrichtung von Parametern und relevanten Berechtigungen für diesen Service zu erhalten.

Configure in Secrets Manager

So konfigurieren Sie Parameterwerte in Secrets Manager

1. Öffnen Sie die Secrets Manager-Konsole.
2. Erstellen Sie ein neues Secret, um Ihre Anmeldeinformationen oder Secrets zu speichern. Anweisungen finden Sie unter Create an AWS Secrets Manager Secret im AWS Secrets Manager Benutzerhandbuch.
3. Kopieren Sie den ARN Ihres Secrets.
4. Fügen Sie die Secrets-Manager-Berechtigungen aus der folgenden Beispielrichtlinie zu der Service-Ausführungsrolle hinzu. Ersetze *<arn:aws:secretsmanager:us-east-*1:123456789000:secret: -1234> durch den ARN deines Secrets. MySecret
5. Fügen Sie Worker-Konfiguration und Konnektor-Anweisungen hinzu.

```
{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "secretsmanager:GetResourcePolicy",
                    "secretsmanager:GetSecretValue",
                    "secretsmanager:DescribeSecret",
                    "secretsmanager:ListSecretVersionIds"
                ],
                "Resource": [
                "<arn:aws:secretsmanager:us-
east-1:123456789000:secret:MySecret-1234>"
                ]
            }
        ]
    }
```

6. Um den Secrets-Manager-Konfigurationsanbieter zu verwenden, kopieren Sie die folgenden Code-Zeilen in das Worker-Konfigurations-Textfeld in Schritt 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. Kopieren Sie für den Secrets-Manager-Konfigurationsanbieter die folgenden Code-Zeilen in die Konnektor-Konfiguration in Schritt 4.

```
#Example implementation for secrets manager variable
database.hostname=${secretsmanager:MSKAuroraDBCredentials:username}
```

```
database.password=${secretsmanager:MSKAuroraDBCredentials:password}
```

Sie können den obigen Schritt auch mit weiteren Konfigurationsanbietern verwenden.

Configure in Systems Manager Parameter Store

So konfigurieren Sie Parameterwerte im Systems Manager Parameter Store

1. Öffnen Sie die Systems Manager-Konsole.
2. Wählen Sie im Navigationsbereich Parameter Store (Parameterspeicher) aus.
3. Erstellen Sie einen neuen Parameter, der im Systems Manager gespeichert werden soll. Anweisungen finden Sie im AWS Systems Manager Benutzerhandbuch unter Erstellen eines Systems Manager Manager-Parameters (Konsole).
4. Kopieren Sie den ARN Ihres Parameters.
5. Fügen Sie die Systems-Manager-Berechtigungen aus der folgenden Beispielrichtlinie zu der Service-Ausführungsrolle hinzu. Ersetzen Sie *<arn:aws:ssm:us-east-1:123456789000:parameter/* > durch den ARN Ihres Parameters. MyParameterName

```
{
        "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. Um den Parameterspeicher-Konfigurationsanbieter zu verwenden, kopieren Sie die folgenden Code-Zeilen in das Worker-Konfigurations-Textfeld in Schritt 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. Kopieren Sie für den Parameterspeicher-Konfigurationsanbieter die folgenden Code-Zeilen in die Konnektor-Konfiguration in Schritt 5.

```
#Example implementation for parameter store variable
schema.history.internal.kafka.bootstrap.servers=
${ssm::MSKBootstrapServerAddress}
```

Sie können den obigen Schritt auch mit weiteren Konfigurationsanbietern bündeln.

Configure in Amazon S3

So konfigurieren Sie Objekte/Dateien in Amazon S3

1. Öffnen Sie die Amazon S3-Konsole.
2. Laden Sie Ihr Objekt in einen Bucket in S3 hoch. Eine Anleitung finden Sie unter Hochladen von Objekten.
3. Kopieren Sie den ARN Ihres Objekts.
4. Fügen Sie die Amazon-S3-Objekt-Leseberechtigungen aus der folgenden Beispielrichtlinie zu der Service-Ausführungsrolle hinzu. Ersetzen Sie *<arn:aws:s3:::MY_S3_BUCKET/path/to/custom-plugin.zip>* durch den ARN Ihres Objekts.

```
{
    "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. Um den Amazon-S3-Konfigurationsanbieter zu verwenden, kopieren Sie die folgenden Code-Zeilen in das Worker-Konfigurations-Textfeld in Schritt 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. Kopieren Sie für den Amazon-S3-Konfigurationsanbieter die folgenden Code-Zeilen in die Konnektor-Konfiguration in Schritt 4.

```
#Example implementation for S3 object

database.ssl.truststore.location = ${s3import:us-west-2:my_cert_bucket/path/to/
trustore_unique_filename.jks}
```

Sie können die obigen zwei Schritte auch mit weiteren Konfigurationsanbietern bündeln.

Schritt 3: Erstellen Sie eine benutzerdefinierte Worker-Konfiguration mit Informationen zu Ihrem Konfigurationsanbieter

1. Wählen Sie im Abschnitt Amazon MSK Connect die Option Worker-Konfigurationen.
2. Wählen Sie Worker-Konfiguration erstellen.
3. Geben Sie `SourceDebeziumCustomConfig` in das Textfeld für den Namen der Worker-Konfiguration ein. Die Beschreibung ist optional.
4. Kopieren Sie den entsprechenden Konfigurations-Code basierend auf den gewünschten Anbietern und fügen Sie ihn in das Textfeld Worker-Konfiguration ein.
5. Dies ist ein Beispiel der Worker-Konfiguration für alle drei Anbieter:

```
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. Wählen Sie Worker-Konfiguration erstellen.

Schritt 4: Den Konnektor erstellen

1. Erstellen Sie einen neuen Konnektor anhand der Anweisungen unter Neuen Konnektor erstellen.
2. Wählen Sie die `custom-plugin.zip`-Datei, die Sie in ??? als Quelle für das benutzerdefinierte Plugin in Ihren S3-Bucket hochgeladen haben.
3. Kopieren Sie den entsprechenden Konfigurations-Code basierend auf den gewünschten Anbietern und fügen Sie ihn in das Feld Konnektor-Konfiguration ein.
4. Dies ist ein Beispiel für die Konnektor-Konfiguration für alle drei Anbieter:

```
#Example implementation for parameter store variable
schema.history.internal.kafka.bootstrap.servers=${ssm::MSKBootstrapServerAddress}
```

```
#Example implementation for secrets manager variable
database.hostname=${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. Wählen Sie Benutzerdefinierte Konfiguration verwenden und wählen Sie eine Option aus der Dropdownliste Worker-Konfiguration aus SourceDebeziumCustomConfig.
6. Folgen Sie den weiteren Schritten aus den Anweisungen unter Konnektor erstellen.

Überlegungen

Beachten Sie bei der Verwendung des MSK-Konfigurationsanbieters mit Amazon MSK Connect Folgendes:

- Weisen Sie der IAM-Service-Ausführungsrolle die entsprechenden Berechtigungen zu, wenn Sie die Konfigurationsanbieter verwenden.
- Definieren Sie die Konfigurationsanbieter in Worker-Konfigurationen und ihre Implementierung in der Konnektor-Konfiguration.
- Vertrauliche Konfigurationswerte können in Konnektor-Protokollen erscheinen, wenn ein Plugin diese Werte nicht als Secret definiert. Kafka Connect behandelt undefinierte Konfigurationswerte genauso wie jeden anderen Klartext-Wert. Weitere Informationen hierzu finden Sie unter Verhindern, dass Secrets in Konnektor-Protokollen erscheinen.
- Standardmäßig startet MSK Connect einen Konnektor häufig neu, wenn der Konnektor einen Konfigurationsanbieter verwendet. Um dieses Neustartverhalten zu deaktivieren, können Sie in der Konnektor-Konfiguration den Wert `config.action.reload` auf `none` festlegen.

IAM-Rollen und -Richtlinien für MSK Connect

Themen

- Service-Ausführungsrolle
- Beispiele für IAM-Richtlinien für MSK Connect
- Serviceübergreifende Confused-Deputy-Prävention

- AWS verwaltete Richtlinien für MSK Connect
- Verwendung von serviceverknüpften Rollen für MSK Connect

Service-Ausführungsrolle

> **Note**
>
> Amazon MSK Connect unterstützt nicht die Verwendung der serviceverknüpften Rolle als Service-Ausführungsrolle. Sie müssen eine separate Service-Ausführungsrolle erstellen. Anweisungen zum Erstellen einer benutzerdefinierten IAM-Rolle finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen für einen AWS Dienst im IAM-Benutzerhandbuch.

Wenn Sie einen Konnektor mit MSK Connect erstellen, müssen Sie eine AWS Identity and Access Management (IAM)-Rolle angeben, die damit verwendet werden soll. Ihre Service-Ausführungsrolle muss die folgende Vertrauensrichtlinie haben, damit MSK Connect sie übernehmen kann. Weitere Informationen zu Bedingungskontextschlüsseln finden Sie unter the section called “Serviceübergreifende Confused-Deputy-Prävention”.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "kafkaconnect.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "Account-ID"
        },
        "ArnLike": {
          "aws:SourceArn": "MSK-Connector-ARN"
        }
      }
    }
  ]
```

```
}
```

Wenn es sich bei dem Amazon-MSK-Cluster, den Sie mit Ihrem Konnektor verwenden möchten, um einen Cluster handelt, der die IAM-Authentifizierung verwendet, müssen Sie der Service-Ausführungsrolle des Konnektors die folgende Berechtigungsrichtlinie hinzufügen. Informationen darüber, wie Sie die UUID Ihres Clusters finden und wie Sie Themen-ARNs erstellen, erhalten Sie unter the section called “Ressourcen”.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:DescribeCluster"
            ],
            "Resource": [
                "cluster-arn"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:ReadData",
                "kafka-cluster:DescribeTopic"
            ],
            "Resource": [
                "ARN of the topic that you want a sink connector to read from"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:WriteData",
                "kafka-cluster:DescribeTopic"
            ],
            "Resource": [
                "ARN of the topic that you want a source connector to write to"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "kafka-cluster:CreateTopic",
                "kafka-cluster:WriteData",
                "kafka-cluster:ReadData",
                "kafka-cluster:DescribeTopic"
            ],
            "Resource": [
                "arn:aws:kafka:region:account-id:topic/cluster-name/cluster-uuid/
__amazon_msk_connect_*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeGroup"
            ],
            "Resource": [
                "arn:aws:kafka:region:account-id:group/cluster-name/cluster-uuid/
__amazon_msk_connect_*",
                "arn:aws:kafka:region:account-id:group/cluster-name/cluster-uuid/
connect-*"
            ]
        }
    ]
}
```

Je nach Art des Connectors müssen Sie der Dienstausführungsrolle möglicherweise auch eine Berechtigungsrichtlinie hinzufügen, die ihr den Zugriff auf Ressourcen ermöglicht. AWS Wenn Ihr Konnektor beispielsweise Daten an einen S3-Bucket senden muss, muss die Service-Ausführungsrolle über eine Berechtigungsrichtlinie verfügen, welche die Erlaubnis erteilt, in diesen Bucket zu schreiben. Zu Testzwecken können Sie eine der vorgefertigten IAM-Richtlinien verwenden, die vollen Zugriff gewähren, z. B. `arn:aws:iam::aws:policy/AmazonS3FullAccess`. Aus Sicherheitsgründen empfehlen wir jedoch, die restriktivste Richtlinie zu verwenden, die es Ihrem Connector ermöglicht, von der AWS Quelle zu lesen oder in die AWS Senke zu schreiben.

Beispiele für IAM-Richtlinien für MSK Connect

Um einem Benutzer ohne Administratorrechte vollen Zugriff auf alle MSK-Connect-Funktionen zu gewähren, fügen Sie der IAM-Rolle des Benutzers eine Richtlinie wie die folgende hinzu.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafkaconnect:*",
                "ec2:CreateNetworkInterface",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:DescribeSecurityGroups",
                "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries",
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
kafkaconnect.amazonaws.com/AWSServiceRoleForKafkaConnect*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "kafkaconnect.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
kafkaconnect.amazonaws.com/AWSServiceRoleForKafkaConnect*"
        },
        {
            "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"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutBucketPolicy",
                "s3:GetBucketPolicy"
            ],
            "Resource": "ARN of the Amazon S3 bucket to which you want MSK Connect to
 deliver logs"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "ARN of the service execution role"
        },
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "ARN of the Amazon S3 object that corresponds to the custom
 plugin that you want to use for creating connectors"
        },
        {
            "Effect": "Allow",
            "Action": "firehose:TagDeliveryStream",
            "Resource": "ARN of the Firehose delivery stream to which you want MSK
 Connect to deliver logs"
        }
    ]
}
```

Serviceübergreifende Confused-Deputy-Prävention

Das Confused-Deputy-Problem ist ein Sicherheitsproblem, bei dem eine juristische Stelle, die nicht über die Berechtigung zum Ausführen einer Aktion verfügt, eine privilegiertere juristische Stelle zwingen kann, die Aktion auszuführen. In AWS, dienstübergreifender Identitätswechsel kann zum Problem des verwirrten Stellvertreters führen. Ein dienstübergreifender Identitätswechsel

kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der aufrufende Service kann manipuliert werden, um seine Berechtigungen zu verwenden, um Aktionen auf die Ressourcen eines anderen Kunden auszuführen, für die er sonst keine Zugriffsberechtigung haben sollte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Wir empfehlen die Verwendung der globalen Bedingungskontext-Schlüssel `aws:SourceArn` und `aws:SourceAccount` in ressourcenbasierten Richtlinien, um die Berechtigungen, die MSK Connect einem anderen Service erteilt, auf die Ressource zu beschränken. Wenn der `aws:SourceArn`-Wert nicht die Konto-ID enthält (z. B. ein Amazon-S3-Bucket-ARN enthält nicht die Konto-ID), müssen Sie beide globale Bedingungskontext-Schlüssel verwenden, um Berechtigungen einzuschränken. Wenn Sie beide globale Bedingungskontextschlüssel verwenden und der `aws:SourceArn`-Wert die Konto-ID enthält, müssen der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert dieselbe Konto-ID verwenden, wenn sie in der gleichen Richtlinienanweisung verwendet wird. Verwenden Sie `aws:SourceArn`, wenn Sie nur eine Ressource mit dem betriebsübergreifenden Zugriff verknüpfen möchten. Verwenden Sie `aws:SourceAccount`, wenn Sie zulassen möchten, dass Ressourcen in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft werden.

Im Fall von MSK Connect muss der Wert von `aws:SourceArn` ein MSK-Konnektor sein.

Der effektivste Weg, um sich vor dem Confused-Deputy-Problem zu schützen, ist die Verwendung des globalen Bedingungskontext-Schlüssels `aws:SourceArn` mit dem vollständigen ARN der Ressource. Wenn Sie den vollständigen ARN der Ressource nicht kennen oder wenn Sie mehrere Ressourcen angeben, verwenden Sie den globalen Bedingungskontext-Schlüssel `aws:SourceArn` mit Platzhaltern (*) für die unbekannten Teile des ARN. Beispielsweise steht *`arn:aws:kafkaconnect:us-east-1:123456789012:connector/*`* für alle Konnektoren, die zu dem Konto mit der ID 123456789012 in der Region USA Ost (Nord-Virginia) gehören.

Das folgende Beispiel zeigt, wie Sie die globalen Bedingungskontext-Schlüssel `aws:SourceArn` und `aws:SourceAccount` in MSK Connect verwenden können, um das Confused-Deputy-Problem zu vermeiden. Ersetzen Sie *`Konto-ID`* und *`MSK-Konnektor-ARN`* durch Ihre Informationen.

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

```
        "Service": " kafkaconnect.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "Account-ID"
        },
        "ArnLike": {
          "aws:SourceArn": "MSK-Connector-ARN"
        }
      }
    }
  ]
}
```

AWS verwaltete Richtlinien für MSK Connect

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet AWS wird. AWS Verwaltete Richtlinien sind so konzipiert, dass sie Berechtigungen für viele gängige Anwendungsfälle bereitstellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie allen AWS Kunden zur Verfügung stehen. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie kundenverwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinie: AmazonMSK ConnectReadOnlyAccess

Diese Richtlinie gewährt dem Benutzer die Berechtigungen, die zum Auflisten und Beschreiben von MSK-Connect-Ressourcen erforderlich sind.

Sie können die `AmazonMSKConnectReadOnlyAccess`-Richtlinie an Ihre IAM-Identitäten anfügen.

```
{
    "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 verwaltete Richtlinie: KafkaConnectServiceRolePolicy

Diese Richtlinie gewährt dem MSK-Connect-Service die Berechtigungen, die zum Erstellen und Verwalten von Netzwerkschnittstellen mit dem Tag `AmazonMSKConnectManaged:true` erforderlich sind. Diese Netzwerkschnittstellen ermöglichen MSK Connect Netzwerkzugriff auf Ressourcen in Ihrer Amazon VPC, wie z. B. einen Apache-Kafka-Cluster oder eine Quelle oder einen Sink.

Sie können keine Verbindungen KafkaConnectServiceRolePolicy zu Ihren IAM-Entitäten herstellen. Diese Richtlinie ist an eine serviceverknüpfte Rolle angehängt, die MSK Connect die Durchführung von Aktionen in Ihrem Namen erlaubt.

```
{
 "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-Updates für AWS verwaltete Richtlinien

Sehen Sie sich Details zu Aktualisierungen der AWS verwalteten Richtlinien für MSK Connect an, seit dieser Dienst begonnen hat, diese Änderungen zu verfolgen.

Änderung	Beschreibung	Datum
MSK Connect hat die schreibgeschützte Richtlinie aktualisiert	MSK Connect hat die ConnectReadOnlyAccess AmazonMSK-Richtlinie aktualisiert, um die Einschrän	13. Oktober 2021

Änderung	Beschreibung	Datum
	kungen bei der Angebotse rstellung aufzuheben.	
MSK Connect hat mit der Nachverfolgung von Änderungen begonnen	MSK Connect begann, Änderungen für seine AWS verwalteten Richtlinien zu verfolgen.	14. September 2021

Verwendung von serviceverknüpften Rollen für MSK Connect

Amazon MSK Connect verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist ein spezieller Typ von IAM-Rolle, die direkt mit MSK Connect verknüpft ist. Dienstbezogene Rollen sind von MSK Connect vordefiniert und enthalten alle Berechtigungen, die der Dienst benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen.

Eine serviceverknüpfte Rolle vereinfacht die Einrichtung von MSK Connect, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. MSK Connect definiert die Berechtigungen seiner serviceverknüpften Rollen. Sofern nicht anders festgelegt, kann nur MSK Connect die Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Informationen zu anderen Services, die serviceverknüpften Rollen unterstützen, finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Serviceverknüpfte Rollenberechtigungen für MSK Connect

MSK Connect verwendet die serviceverknüpfte Rolle mit dem Namen AWSServiceRoleForKafkaConnect— Erlaubt Amazon MSK Connect, in Ihrem Namen auf Amazon-Ressourcen zuzugreifen.

Die AWSServiceRoleForKafkaConnect dienstbezogene Rolle vertraut darauf, dass der `kafkaconnect.amazonaws.com` Service die Rolle übernimmt.

Weitere Informationen über die Berechtigungsrichtlinie, die die Rolle verwendet, finden Sie unter the section called “KafkaConnectServiceRolePolicy”.

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter serviceverknüpfte Rollenberechtigungen im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für MSK Connect

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie einen Connector in der AWS Management Console, der oder der AWS API erstellen AWS CLI, erstellt MSK Connect die serviceverknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie einen Konnektor erstellen, erstellt MSK Connect wieder die serviceverknüpfte Rolle für Sie.

Bearbeiten einer serviceverknüpften Rolle für MSK Connect

MSK Connect erlaubt es Ihnen nicht, die AWSServiceRoleForKafkaConnect dienstverknüpfte Rolle zu bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung einer serviceverknüpften Rolle nicht bearbeitet werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für MSK Connect

Sie können die IAM-Konsole, die AWS CLI oder die AWS API verwenden, um die serviceverknüpfte Rolle manuell zu löschen. Dazu müssen Sie zunächst alle MSK-Connect-Konnektoren manuell löschen und dann können Sie die Rolle manuell löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Leitfaden.

Unterstützte Regionen für serviceverknüpfte MSK-Connect-Rollen

MSK Connect unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter AWS Regionen und Endpunkte.

Aktivieren des Internetzugangs für Amazon MSK Connect

Wenn Ihr Connector für Amazon MSK Connect Zugriff auf das Internet benötigt, empfehlen wir Ihnen, die folgenden Amazon Virtual Private Cloud (VPC-) Einstellungen zu verwenden, um diesen Zugriff zu aktivieren.

- Konfigurieren Sie Ihren Konnektor mit privaten Subnetzen.
- Erstellen Sie ein öffentliches NAT-Gateway oder eine NAT-Instance für Ihre VPC in einem öffentlichen Subnetz. Weitere Informationen finden Sie auf der Seite Verbinden von Subnetzen mit dem Internet oder anderen VPCs mithilfe von NAT-Geräten im Amazon Virtual Private CloudBenutzerhandbuch.
- Erlauben Sie ausgehenden Datenverkehr von Ihren privaten Subnetzen zu Ihrem NAT-Gateway oder Ihrer NAT-Instance.

Einrichtung eines NAT-Gateways für Amazon MSK Connect

In den folgenden Schritten wird gezeigt, wie Sie ein NAT-Gateway einrichten, um den Internetzugang für einen Konnektor zu ermöglichen. Sie müssen diese Schritte ausführen, bevor Sie einen Konnektor in einem privaten Subnetz erstellen.

Voraussetzungen

Stellen Sie sicher, dass Sie über Folgendes verfügen.

- Die ID der Amazon Virtual Private Cloud (VPC), die Ihrem Cluster zugeordnet ist. Zum Beispiel vpc-123456ab.
- Die IDs der privaten Subnetze in Ihrer VPC. Zum Beispiel subnet-a1b2c3de, subnet-f4g5h6ij usw. Sie müssen Ihren Konnektor mit privaten Subnetzen konfigurieren.

Internetzugang für Ihren Konnektor aktivieren

1. Öffnen Sie die Amazon Virtual Private Cloud Konsole unter https://console.aws.amazon.com/vpc/.
2. Erstellen Sie ein öffentliches Subnetz für Ihr NAT-Gateway mit einem aussagekräftigen Namen und notieren Sie sich die Subnetz-ID. Detaillierte Anweisungen finden Sie unter Erstellen eines Subnetzes in Ihrer VPC.

3. Erstellen Sie ein Internet-Gateway, damit Ihre VPC mit dem Internet kommunizieren kann, und notieren Sie sich die Gateway-ID. Anfügen eines Internet-Gateways zu Ihrer VPC. Anweisungen finden Sie unter Erstellen und Anfügen eines Internet-Gateway.
4. Stellen Sie ein öffentliches NAT-Gateway bereit, damit Hosts in Ihren privaten Subnetzen Ihr öffentliches Subnetz erreichen können. Wenn Sie das NAT-Gateway erstellen, wählen Sie das öffentliche Subnetz aus, das Sie zuvor erstellt haben. Detaillierte Anweisungen finden Sie unter Erstellen eines NAT-Gateway.
5. Konfigurieren Sie Ihre Routing-Tabellen. Sie benötigen insgesamt zwei Routing-Tabellen, um diese Einrichtung abzuschließen. Sie sollten bereits über eine Haupt-Routing-Tabelle verfügen, die automatisch zur gleichen Zeit wie Ihre VPC erstellt wurde. In diesem Schritt erstellen Sie eine zusätzliche Routing-Tabelle für Ihr öffentliches Subnetz.

   a. Verwenden Sie die folgenden Einstellungen, um die Haupt-Routing-Tabelle Ihrer VPC so zu ändern, dass Ihre privaten Subnetze den Verkehr an Ihr NAT-Gateway weiterleiten. Anweisungen finden Sie im Benutzerhandbuch für Amazon Virtual Private Cloud unter Arbeiten mit Routing-Tabellen.

      Private MSKC-Routing-Tabelle

Eigenschaft	Wert
Namens-Tag	Es wird empfohlen, dieser Routing-Tabelle einen aussagekräftigen Namen zu geben, damit Sie sie leichter identifizieren können. Zum Beispiel Private MSKC.
Assoziierte Subnetze	Ihre privaten Subnetze
Eine Route, um den Internetzugang für MSK Connect zu aktivieren	• Routenziel: 0.0.0.0/0 • Ziel: Ihre NAT-Gateway-ID. Zum Beispiel nat-12a345bc6789efg1h.
Eine lokale Route für internen Datenverk ehr	• Ziel: 10.0.0.0/16. Dieser Wert kann je nach CIDR-Block Ihrer VPC unterschi edlich sein. • Ziel: Lokal

b. Folgen Sie den Anweisungen unter Erstellen einer benutzerdefinierten Routing-Tabelle, um eine Routing-Tabelle für Ihr öffentliches Subnetz zu erstellen. Geben Sie beim Erstellen der Tabelle einen aussagekräftigen Namen in das Feld Namens-Tag ein, damit Sie leichter erkennen können, mit welchem Subnetz die Tabelle verknüpft ist. Zum Beispiel Public MSKC.

c. Konfigurieren Sie Ihre Public MSKC-Routing-Tabelle mit den folgenden Einstellungen.

Eigenschaft	Wert
Namens-Tag	Public MSKC oder ein anderer beschreib ender Name, den Sie wählen
Assoziierte Subnetze	Ihr öffentliches Subnetz mit NAT-Gateway
Eine Route, um den Internetzugang für MSK Connect zu aktivieren	• Ziel: 0.0.0.0/0 • Ziel: Ihre Internet-Gateway-ID. Zum Beispiel igw-1a234bc5.
Eine lokale Route für internen Datenverk ehr	• Ziel: 10.0.0.0/16. Dieser Wert kann je nach CIDR-Block Ihrer VPC unterschi edlich sein. • Ziel: Lokal

Private DNS-Hostnamen

Mit der Unterstützung für private DNS-Hostnamen in MSK Connect können Sie Konnektoren so konfigurieren, dass sie auf öffentliche oder private Domainnamen verweisen. Die Unterstützung hängt von den DNS-Servern ab, die im DHCP-Optionssatz der VPC angegeben sind.

Ein DHCP-Optionssatz ist eine Gruppe von Netzwerkkonfigurationen, mit deren Hilfe EC2-Instances in einer VPC über Ihr virtuelles Netzwerk kommunizieren. Jede VPC weist einen standardmäßigen DHCP-Optionssatz auf. Sie können jedoch einen benutzerdefinierten DHCP-Optionssatz erstellen, etwa wenn die Instances in Ihrer VPC anstelle des Amazon-DNS-Servers einen anderen DNS-Server für die Auflösung von Domainnamen verwenden sollen. Siehe DHCP-Optionssätze in Amazon VPC.

Bevor die Private DNS-Auflösungskapazität bzw. -Feature in MSK Connect enthalten war, verwendeten Konnektoren die Service-VPC-DNS-Resolver für DNS-Abfragen von einem Kunden-

Konnektor. Konnektoren verwendeten nicht die DNS-Server, die in den VPC-DHCP-Optionssätzen des Kunden für die DNS-Auflösung definiert sind.

Konnektoren konnten nur in Konnektor-Konfigurationen oder Plugins des Kunden, die öffentlich auflösbar waren, auf Hostnamen verweisen. Sie konnten keine privaten Hostnamen auflösen, die in einer privat gehosteten Zone definiert waren, oder DNS-Server in einem anderen Kundennetzwerk verwenden.

Ohne Private DNS konnten Kunden, die sich dafür entschieden hatten, ihre Datenbanken, Data Warehouses und Systeme wie den Secrets Manager in ihrer eigenen VPC für das Internet unzugänglich zu machen, nicht mit MSK-Konnektoren arbeiten. Kunden verwenden häufig private DNS-Hostnamen, um die Sicherheitsvorkehrungen des Unternehmens einzuhalten.

Themen

Konfiguration eines VPC-DHCP-Optionssatzes für Ihren Konnektor

Konnektoren verwenden automatisch die DNS-Server, die in ihrem VPC-DHCP-Optionssatz definiert sind, wenn der Konnektor erstellt wird. Bevor Sie einen Konnektor erstellen, stellen Sie sicher, dass Sie den VPC-DHCP-Optionssatz für die DNS-Hostnamen-Auflösungsanforderungen Ihres Konnektors konfigurieren.

Konnektoren, die Sie erstellt haben, bevor das Private-DNS-Hostname-Feature in MSK Connect verfügbar war, verwenden weiterhin die vorherige DNS-Auflösungskonfiguration, ohne dass Änderungen erforderlich sind.

Wenn Sie in Ihrem Konnektor nur eine öffentlich auflösbare DNS-Hostname-Auflösung benötigen, empfehlen wir zur einfacheren Einrichtung, bei der Erstellung des Konnektors die Standard-VPC Ihres Kontos zu verwenden. Weitere Informationen zum von Amazon bereitgestellten DNS-Server oder Amazon Route 53 Resolver finden Sie unter Amazon DNS Server im Amazon-VPC-Benutzerhandbuch.

Wenn Sie private DNS-Hostnamen auflösen müssen, stellen Sie sicher, dass der DHCP-Optionssatz der VPC, die bei der Erstellung des Konnektors übergeben wird, korrekt konfiguriert

ist. Weitere Informationen finden Sie unter Arbeiten mit DHCP-Optionssätzen im Amazon-VPC-Benutzerhandbuch.

Wenn Sie einen DHCP-Optionssatz für die Auflösung privater DNS-Hostnamen konfigurieren, stellen Sie sicher, dass der Konnektor die benutzerdefinierten DNS-Server erreichen kann, die Sie im DHCP-Optionssatz konfigurieren. Andernfalls schlägt die Erstellung des Konnektors fehl.

Nachdem Sie den VPC-DHCP-Optionssatz angepasst haben, verwenden Konnektoren, die anschließend in dieser VPC erstellt wurden, die DNS-Server, die Sie im Optionssatz angegeben haben. Wenn Sie den Optionssatz ändern, nachdem Sie einen Konnektor erstellt haben, übernimmt der Konnektor innerhalb weniger Minuten die Einstellungen im neuen Optionssatz.

DNS-Attribute für Ihre VPC

Stellen Sie sicher, dass Sie die VPC-DNS-Attribute korrekt konfiguriert haben, wie unter DNS-Attribute in Ihrer VPC und DNS-Hostnamen im Amazon-VPC-Benutzerhandbuch beschrieben.

Unter Auflösen von DNS-Abfragen zwischen VPCs und Ihrem Netzwerk im Entwicklerhandbuch für Amazon Route 53 finden Sie Informationen zur Verwendung von Resolver-Endpunkten für eingehende und ausgehende Verbindungen, um andere Netzwerke mit Ihrer VPC zu verbinden und mit Ihrem Konnektor zu arbeiten.

Fehlerbehandlung

Dieser Abschnitt beschreibt mögliche Fehler bei der Konnektor-Erstellung im Zusammenhang mit der DNS-Auflösung und empfohlene Maßnahmen zur Behebung der Probleme.

Fehler	Vorgeschlagene Aktion
Die Konnektor-Erstellung schlägt fehl, wenn eine DNS-Auflösungsabfrage fehlschlägt oder wenn DNS-Server vom Konnektor aus nicht erreichbar sind.	Sie können Fehler bei der Connectorerstellung aufgrund erfolgloser DNS-Auflösungsabfragen in Ihren CloudWatch Protokollen sehen, wenn Sie diese Protokolle für Ihren Connector konfiguriert haben. Überprüfen Sie die DNS-Serverkonfigurationen und stellen Sie die Netzwerkkonnektivität zu den DNS-Servern vom Konnektor aus sicher.

Fehler	Vorgeschlagene Aktion
Wenn Sie die DNS-Serverkonfiguration in Ihrem VPC-DHCP-Optionssatz ändern, während ein Konnektor läuft, können DNS-Auflösungsabfragen vom Konnektor fehlschlagen. Wenn die DNS-Auflösung fehlschlägt, können einige der Konnektor-Aufgaben in den Status „Fehlgeschlagen“ übergehen.	Wenn Sie diese Protokolle für Ihren Connector konfiguriert haben, können Sie Fehler bei der Connectorerstellung aufgrund erfolgloser DNS-Auflösungsabfragen in Ihren CloudWatch Protokollen sehen. Die fehlgeschlagenen Aufgaben sollten automatisch neu gestartet werden, damit der Konnektor wieder betriebsbereit ist. Geschieht dies nicht, können Sie sich an den Support wenden, um die fehlgeschlagenen Aufgaben für den jeweiligen Konnektor neu zu starten, oder Sie können den Konnektor neu erstellen.

Protokollierung für MSK Connect

MSK Connect kann Protokollereignisse schreiben, die Sie zum Debuggen Ihres Konnektors verwenden können. Wenn Sie einen Konnektor erstellen, können Sie null oder mehr der folgenden Protokollziele angeben:

- Amazon CloudWatch Logs: Sie geben die Protokollgruppe an, an die MSK Connect die Protokollereignisse Ihres Connectors senden soll. Informationen zum Erstellen einer Protokollgruppe finden Sie unter Erstellen einer Protokollgruppe im CloudWatch Logs-Benutzerhandbuch.
- Amazon S3: Sie geben den S3-Bucket an, an den MSK Connect die Protokollereignisse Ihres Konnektors senden soll. Weitere Informationen zum Erstellen eines S3-Buckets finden Sie unter Erstellen eines Buckets im Benutzerhandbuch für Amazon S3.
- Amazon Data Firehose: Sie geben den Lieferstream an, an den MSK Connect die Protokollereignisse Ihres Connectors senden soll. Informationen zum Erstellen eines Lieferdatenstroms finden Sie unter Erstellen eines Amazon Data Firehose-Lieferdatenstroms im Firehose-Benutzerhandbuch.

Weitere Informationen zum Einrichten der Protokollierung finden Sie unter Aktivieren der Protokollierung von AWS -Services im Amazon CloudWatch Logs -Benutzerhandbuch.

MSK Connect gibt die folgenden Arten von Protokollereignissen aus:

Level	Beschreibung
`INFO`	Interessante Laufzeitereignisse beim Startup und Herunterfahren.
`WARN`	Laufzeitsituationen, die keine Fehler sind, aber unerwünscht oder unerwartet sind.
`FATAL`	Schwerwiegende Fehler, die zu einer vorzeitig en Beendigung führen.
`ERROR`	Unerwartete Bedingungen und Laufzeitfehler, die nicht schwerwiegend sind.

Im Folgenden finden Sie ein Beispiel für ein Protokollereignis, das an Logs gesendet CloudWatch wurde:

```
[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)
```

Verhindern, dass Secrets in Konnektor-Protokollen erscheinen

> **Note**
>
> Vertrauliche Konfigurationswerte können in Konnektor-Protokollen erscheinen, wenn ein Plugin diese Werte nicht als Secret definiert. Kafka Connect behandelt undefinierte Konfigurationswerte genauso wie jeden anderen Klartext-Wert.

Wenn Ihr Plugin eine Eigenschaft als Secret definiert, redigiert Kafka Connect den Wert der Eigenschaft aus den Konnektor-Protokollen. Die folgenden Konnektor-Protokolle zeigen beispielsweise, dass, wenn ein Plugin `aws.secret.key` als `PASSWORD` Typ definiert, sein Wert durch **`[hidden]`** ersetzt wird.

```
   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)
```

Um zu verhindern, dass Secrets in Konnektor-Protokolldateien auftauchen, muss ein Plugin-Entwickler die Enum-Konstante `ConfigDef.Type.PASSWORD` von Kafka Connect verwenden, um sensible Eigenschaften zu definieren. Wenn eine Eigenschaft vom Typ `ConfigDef.Type.PASSWORD` ist, schließt Kafka Connect ihren Wert aus den Konnektor-Protokollen aus, auch wenn der Wert als Klartext gesendet wird.

Überwachen von MSK Connect

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von MSK Connect und Ihren anderen AWS Lösungen. Amazon CloudWatch überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie laufen, AWS in Echtzeit. Sie können Kennzahlen erfassen und verfolgen, benutzerdefinierte Dashboards erstellen und Alarme festlegen, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen von Ihnen festgelegten Schwellenwert erreicht. Sie können beispielsweise die CPU-Auslastung oder andere Messwerte Ihres Connectors CloudWatch verfolgen, sodass Sie dessen Kapazität bei Bedarf erhöhen können. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

Die folgende Tabelle zeigt die Metriken, an die MSK Connect CloudWatch unter der `ConnectorName` Dimension sendet. MSK Connect liefert diese Metriken standardmäßig und ohne zusätzliche Kosten. CloudWatch speichert diese Metriken 15 Monate lang, sodass Sie auf historische Informationen zugreifen und sich einen besseren Überblick über die Leistung Ihrer Konnektoren verschaffen können. Sie können auch Alarme einrichten, die auf bestimmte Grenzwerte achten und Benachrichtigungen senden oder Aktivitäten auslösen, wenn diese Grenzwerte erreicht werden. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

MSK-Connect-Metriken

Metrikname	Beschreibung
`BytesInPerSec`	Die Gesamtanzahl der vom Konnektor empfangenen Bytes.
`BytesOutPerSec`	Die Gesamtanzahl der vom Konnektor bereitgestellten Bytes.
`CpuUtilization`	Der prozentuale Anteil des CPU-Verbrauchs nach System und Benutzer.
`ErroredTaskCount`	Die Anzahl von fehlerhaften Aufgaben.
`MemoryUtilization`	Der Prozentsatz des Gesamtspeichers auf einer Worker-Instance, nicht nur der Heap-Speicher der Java Virtual Machine (JVM), der derzeit verwendet wird. JVM gibt normalerw eise keinen Speicher an das Betriebssystem zurück. Daher beginnt die JVM-Heap-Größe (MemoryUtilization) normalerweise mit einer minimalen Heap-Größe, die schrittweise auf ein stabiles Maximum von etwa 80-90% ansteigt. Die JVM-Heap-Nutzung kann zunehmen oder abnehmen, wenn sich die tatsächliche Speicherauslastung des Konnektors ändert.
`RebalanceCompletedTotal`	Die Gesamtzahl der von diesem Konnektor durchgeführten Neuausgleichungen.
`RebalanceTimeAvg`	Die durchschnittliche Zeit in Millisekunden, die der Konnektor für den Neuausgleich benötigt.
`RebalanceTimeMax`	Die maximale Zeit in Millisekunden, die der Konnektor für den Neuausgleich benötigt.
`RebalanceTimeSinceLast`	Die Zeit in Millisekunden, seit dieser Konnektor den letzten Neuausgleich abgeschlossen hat.

Metrikname	Beschreibung
RunningTaskCount	Die Anzahl der Aufgaben, die im Konnektor ausgeführt werden.
SinkRecordReadRate	Die durchschnittliche Anzahl der pro Sekunde aus dem Apache-Kafka- oder Amazon-MSK-Cluster gelesenen Datensätze.
SinkRecordSendRate	Die durchschnittliche Anzahl von Datensätzen pro Sekunde, die von den Transformationen ausgegeben und an das Ziel gesendet werden. Diese Zahl beinhaltet keine gefilterten Datensätze.
SourceRecordPollRate	Die durchschnittliche Anzahl der pro Sekunde erstellten oder abgefragten Datensätze.
SourceRecordWriteRate	Die durchschnittliche Anzahl pro Sekunde der von den Transformationen ausgegebenen und in den Apache-Kafka- oder Amazon-MSK-Cluster geschriebenen Datensätze.
TaskStartupAttemptsTotal	Die Gesamtzahl der Aufgaben-Startups, die der Konnektor versucht hat. Sie können diese Metrik verwenden, um Anomalien bei Startup-Versuchen von Aufgaben zu identifizieren.
TaskStartupSuccessPercentage	Der durchschnittliche Prozentsatz erfolgreicher Aufgaben-Startups für den Konnektor. Sie können diese Metrik verwenden, um Anomalien bei Startup-Versuchen von Aufgaben zu identifizieren.
WorkerCount	Die Anzahl der Worker, die dem Konnektor zugewiesen sind.

Beispiele

Dieser Abschnitt enthält Beispiele, die Ihnen bei der Einrichtung von Amazon-MSK-Connect-Ressourcen wie gängigen Konnektoren und Konfigurationsanbietern von Drittanbietern helfen sollen.

Themen

Amazon S3 Sink Connector

Dieses Beispiel zeigt, wie der Amazon S3 S3-Sink-Connector von Confluent verwendet wird und wie ein Amazon S3 S3-Sink-Connector in MSK Connect erstellt wird. AWS CLI

1. Kopieren Sie den folgenden JSON-Code und fügen Sie diesen in eine neue Datei ein. Ersetzen Sie die Platzhalterzeichenfolgen durch Werte, die der Bootstrap-Server-Verbindungszeichenfolge Ihres Amazon-MSK-Clusters und den Subnetz- und Sicherheitsgruppen-IDs des Clusters entsprechen. Informationen zum Einrichten einer Service-Ausführungsrolle finden Sie unter the section called “IAM-Rollen und -Richtlinien”.

```
{
    "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": "my-test-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. Führen Sie den folgenden AWS CLI Befehl in dem Ordner aus, in dem Sie die JSON-Datei im vorherigen Schritt gespeichert haben.

```
aws kafkaconnect create-connector --cli-input-json file://connector-info.json
```

Das Folgende ist ein Beispiel für die Ausgabe, die Sie erhalten, wenn Sie den Befehl erfolgreich ausführen.

```
{
    "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"
```

```
}
```

Debezium-Quell-Konnektor mit Konfigurationsanbieter

Dieses Beispiel zeigt, wie das Debezium-MySQL-Konnektor-Plugin mit einer MySQL-kompatiblen Amazon-Aurora-Datenbank als Quelle verwendet wird. In diesem Beispiel haben wir auch den Open-Source AWS Secrets Manager Config Provider für die Externalisierung von Datenbank-Anmeldeinformationen in AWS Secrets Manager eingerichtet. Weitere Informationen zu Konfigurationsanbietern finden Sie unter Externalisierung vertraulicher Informationen mithilfe von Konfigurationsanbietern.

> **Important**
>
> Das Debezium-MySQL-Konnektor-Plugin unterstützt nur eine Aufgabe und funktioniert nicht mit dem automatisch skalierten Kapazitätsmodus für Amazon MSK Connect. Sie sollten stattdessen den Modus Bereitgestellte Kapazität verwenden und in der Konnektor-Konfiguration den Wert `workerCount` auf Eins festlegen. Weitere Informationen zu den Kapazitätsmodi für MSK Connect finden Sie unter Kapazität des Konnektors.

Bevor Sie beginnen

Ihr Connector muss auf das Internet zugreifen können, damit er mit Diensten interagieren kann AWS Secrets Manager , die sich beispielsweise außerhalb Ihres befinden Amazon Virtual Private Cloud. Die Schritte in diesem Abschnitt helfen Ihnen dabei, die folgenden Aufgaben auszuführen, um den Internetzugang zu aktivieren.

- Richten Sie ein öffentliches Subnetz ein, das ein NAT-Gateway hostet und den Datenverkehr an ein Internet-Gateway in Ihrer VPC weiterleitet.
- Erstellen Sie eine Standardroute, die Ihren privaten Subnetzverkehr an Ihr NAT-Gateway weiterleitet.

Weitere Informationen finden Sie unter Aktivieren des Internetzugangs für Amazon MSK Connect.

Voraussetzungen

Bevor Sie den Internetzugang aktivieren können, benötigen Sie die folgenden Elemente:

- Die ID der Amazon Virtual Private Cloud (VPC), die Ihrem Cluster zugeordnet ist. Zum Beispiel vpc-123456ab.
- Die IDs der privaten Subnetze in Ihrer VPC. Zum Beispiel subnet-a1b2c3de, subnet-f4g5h6ij usw. Sie müssen Ihren Konnektor mit privaten Subnetzen konfigurieren.

Internetzugang für Ihren Konnektor aktivieren

1. Öffnen Sie die Amazon Virtual Private Cloud Konsole unter https://console.aws.amazon.com/vpc/.
2. Erstellen Sie ein öffentliches Subnetz für Ihr NAT-Gateway mit einem aussagekräftigen Namen und notieren Sie sich die Subnetz-ID. Detaillierte Anweisungen finden Sie unter Erstellen eines Subnetzes in Ihrer VPC.
3. Erstellen Sie ein Internet-Gateway, damit Ihre VPC mit dem Internet kommunizieren kann, und notieren Sie sich die Gateway-ID. Anfügen eines Internet-Gateways zu Ihrer VPC. Anweisungen finden Sie unter Erstellen und Anfügen eines Internet-Gateway.
4. Stellen Sie ein öffentliches NAT-Gateway bereit, damit Hosts in Ihren privaten Subnetzen Ihr öffentliches Subnetz erreichen können. Wenn Sie das NAT-Gateway erstellen, wählen Sie das öffentliche Subnetz aus, das Sie zuvor erstellt haben. Detaillierte Anweisungen finden Sie unter Erstellen eines NAT-Gateway.
5. Konfigurieren Sie Ihre Routing-Tabellen. Sie benötigen insgesamt zwei Routing-Tabellen, um diese Einrichtung abzuschließen. Sie sollten bereits über eine Haupt-Routing-Tabelle verfügen, die automatisch zur gleichen Zeit wie Ihre VPC erstellt wurde. In diesem Schritt erstellen Sie eine zusätzliche Routing-Tabelle für Ihr öffentliches Subnetz.

   a. Verwenden Sie die folgenden Einstellungen, um die Haupt-Routing-Tabelle Ihrer VPC so zu ändern, dass Ihre privaten Subnetze den Verkehr an Ihr NAT-Gateway weiterleiten. Anweisungen finden Sie im Benutzerhandbuch für Amazon Virtual Private Cloud unter Arbeiten mit Routing-Tabellen.

      Private MSKC-Routing-Tabelle

Eigenschaft	Wert
Namens-Tag	Es wird empfohlen, dieser Routing-Tabelle einen aussagekräftigen Namen zu geben,

Eigenschaft	Wert
	damit Sie sie leichter identifizieren können. Zum Beispiel Private MSKC.
Assoziierte Subnetze	Ihre privaten Subnetze
Eine Route, um den Internetzugang für MSK Connect zu aktivieren	• Routenziel: 0.0.0.0/0 • Ziel: Ihre NAT-Gateway-ID. Zum Beispiel nat-12a345bc6789efg1h.
Eine lokale Route für internen Datenverkehr	• Ziel: 10.0.0.0/16. Dieser Wert kann je nach CIDR-Block Ihrer VPC unterschiedlich sein. • Ziel: Lokal

b. Folgen Sie den Anweisungen unter Erstellen einer benutzerdefinierten Routing-Tabelle, um eine Routing-Tabelle für Ihr öffentliches Subnetz zu erstellen. Geben Sie beim Erstellen der Tabelle einen aussagekräftigen Namen in das Feld Namens-Tag ein, damit Sie leichter erkennen können, mit welchem Subnetz die Tabelle verknüpft ist. Zum Beispiel Public MSKC.

c. Konfigurieren Sie Ihre Public MSKC-Routing-Tabelle mit den folgenden Einstellungen.

Eigenschaft	Wert
Namens-Tag	Public MSKC oder ein anderer beschreibender Name, den Sie wählen
Assoziierte Subnetze	Ihr öffentliches Subnetz mit NAT-Gateway
Eine Route, um den Internetzugang für MSK Connect zu aktivieren	• Ziel: 0.0.0.0/0 • Ziel: Ihre Internet-Gateway-ID. Zum Beispiel igw-1a234bc5.
Eine lokale Route für internen Datenverkehr	• Ziel: 10.0.0.0/16. Dieser Wert kann je nach CIDR-Block Ihrer VPC unterschiedlich sein. • Ziel: Lokal

Nachdem Sie den Internetzugang für Amazon MSK Connect aktiviert haben, sind Sie bereit, einen Konnektor zu erstellen.

Erstellen eines Debezium-Quell-Konnektors

1. Ein benutzerdefiniertes Plugin erstellen

   a. Laden Sie das MySQL-Konnektor-Plugin für die neueste stabile Version von der Debezium-Webseite herunter. Notieren Sie sich die Debezium-Release-Version, die Sie herunterladen (Version 2.x oder die ältere Serie 1.x). Später in diesem Verfahren werden Sie einen Konnektor erstellen, der auf Ihrer Debezium-Version basiert.

   b. Laden Sie den AWS Secrets Manager Config Provider herunter und extrahieren Sie ihn.

   c. Platzieren Sie die folgenden Archive in das gleiche Verzeichnis:

      - Den Ordner `debezium-connector-mysql`
      - Den Ordner `jcusten-border-kafka-config-provider-aws-0.1.1`

   d. Komprimieren Sie das Verzeichnis, das Sie im vorherigen Schritt erstellt haben, in eine ZIP-Datei und laden Sie die ZIP-Datei dann in einen S3-Bucket hoch. Eine Anleitung finden Sie unter Hochladen von Objekten im Amazon-S3-Benutzerhandbuch.

   e. Kopieren Sie den folgenden JSON-Code und fügen Sie diesen in eine Datei ein. z. B. `debezium-source-custom-plugin.json`. Ersetzen Sie *< example-custom-plugin-name >* durch den Namen, den das Plugin haben soll, *< arn-of-your-s 3-bucket>* durch den ARN des S3-Buckets, in den Sie die ZIP-Datei hochgeladen haben, und *<file-key-of-ZIP-object>* durch den Dateischlüssel des ZIP-Objekts, das Sie auf S3 hochgeladen haben.

```
{
    "name": "<example-custom-plugin-name>",
    "contentType": "ZIP",
    "location": {
        "s3Location": {
            "bucketArn": "<arn-of-your-s3-bucket>",
            "fileKey": "<file-key-of-ZIP-object>"
        }
    }
}
```

   f. Führen Sie den folgenden AWS CLI Befehl in dem Ordner aus, in dem Sie die JSON-Datei gespeichert haben, um ein Plugin zu erstellen.

```
aws kafkaconnect create-custom-plugin --cli-input-json file://<debezium-source-custom-plugin.json>
```

Die Ausgabe sollte in etwa wie folgt aussehen.

```
{
    "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. Führen Sie den folgenden Befehl aus, um den Plugin-Status zu überprüfen. Der Status sollte von `CREATING` zu `ACTIVE` wechseln. Ersetzen Sie den ARN-Platzhalter durch den ARN, den Sie in der Ausgabe des vorherigen Befehls erhalten haben.

```
aws kafkaconnect describe-custom-plugin --custom-plugin-arn "<arn-of-your-custom-plugin>"
```

2. Konfigurieren AWS Secrets Manager und erstellen Sie ein Geheimnis für Ihre Datenbankanmeldedaten

   a. Öffnen Sie die Secrets-Manager-Konsole unterhttps://console.aws.amazon.com/secretsmanager/.

   b. Erstellen Sie ein neues Secret, um Ihre Datenbank-Anmeldeinformationen zu speichern. Anweisungen finden Sie unter Ein Secret erstellen im Benutzerhandbuch für AWS Secrets Manager.

   c. Kopieren Sie den ARN Ihres Secrets.

   d. Fügen Sie die Secrets-Manager-Berechtigungen aus der folgenden Beispielrichtlinie zu der Service-Ausführungsrolle hinzu. Ersetze *<arn:aws:secretsmanager:us-east-*1:123456789000:secret: -1234> durch den ARN deines Secrets. MySecret

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

```
        "secretsmanager:GetResourcePolicy",
        "secretsmanager:GetSecretValue",
        "secretsmanager:DescribeSecret",
        "secretsmanager:ListSecretVersionIds"
      ],
      "Resource": [
        "<arn:aws:secretsmanager:us-east-1:123456789000:secret:MySecret-1234>"
      ]
    }
  ]
}
```

Informationen zum Verwalten von IAM-Berechtigungen finden Sie unter Hinzufügen und Entfernen von IAM-Identitätsberechtigungen im IAM-Benutzerhandbuch.

3. Eine benutzerdefinierte Worker-Konfiguration mit Informationen zu Ihrem Konfigurationsanbieter erstellen

   a. Kopieren Sie die folgenden Eigenschaften der Worker-Konfiguration in eine Datei und ersetzen Sie die Platzhalterzeichenfolgen durch Werte, die Ihrem Szenario entsprechen. Weitere Informationen zu den Konfigurationseigenschaften für den AWS Secrets Manager Config Provider finden Sie SecretsManagerConfigProviderin der Dokumentation des Plugins.

   ```
   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. Führen Sie den folgenden AWS CLI Befehl aus, um Ihre benutzerdefinierte Worker-Konfiguration zu erstellen.

   Ersetzen Sie die folgenden Werte:

   - *< my-worker-config-name >* — ein beschreibender Name für Ihre benutzerdefinierte Worker-Konfiguration
   - *< encoded-properties-file-content -string>* — eine Base64-kodierte Version der Klartext-Eigenschaften, die Sie im vorherigen Schritt kopiert haben

```
aws kafkaconnect create-worker-configuration --name <my-worker-config-name> --
properties-file-content <encoded-properties-file-content-string>
```

4. Erstellen eines Konnektors

   a. Kopieren Sie den folgenden JSON-Code, der Ihrer Debezium-Version (2.x oder 1.x) entspricht, und fügen Sie ihn in eine neue Datei ein. Ersetzen Sie die Zeichenfolge *<placeholder>* durch Werte, die Ihrem Szenario entsprechen. Informationen zum Einrichten einer Service-Ausführungsrolle finden Sie unter the section called “IAM-Rollen und -Richtlinien”.

   Beachten Sie, dass die Konfiguration Variablen wie `${secretManager:MySecret-1234:dbusername}` anstelle von Klartext verwendet, um Datenbank-Anmeldeinformationen anzugeben. Ersetzen Sie *MySecret-1234* durch den Namen Ihres Secrets und geben Sie dann den Namen des Schlüssels an, den Sie abrufen möchten. Sie müssen auch *<arn-of-config-provider-worker-configuration>* durch den ARN Ihrer benutzerdefinierten Worker-Konfiguration ersetzen.

   Debezium 2.x

   Kopieren Sie für Debezium-2.x-Versionen den folgenden JSON-Code und fügen Sie ihn in eine neue Datei ein. Ersetzen Sie die *<placeholder>*-Zeichenfolgen durch Werte, die Ihrem Szenario entsprechen.

```
{
 "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

Kopieren Sie für Debezium-1.x-Versionen den folgenden JSON-Code und fügen Sie ihn in eine neue Datei ein. Ersetzen Sie die *<placeholder>*-Zeichenfolgen durch Werte, die Ihrem Szenario entsprechen.

```
{
 "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. Führen Sie den folgenden AWS CLI Befehl in dem Ordner aus, in dem Sie die JSON-Datei im vorherigen Schritt gespeichert haben.

```
aws kafkaconnect create-connector --cli-input-json file://connector-info.json
```

Das Folgende ist ein Beispiel für die Ausgabe, die Sie erhalten, wenn Sie den Befehl erfolgreich ausführen.

```
{
    "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"
}
```

Ein Beispiel für einen Debezium-Konnektor mit detaillierten Schritten finden Sie unter Einführung in Amazon MSK Connect – Streamen Sie Daten mithilfe von verwalteten Konnektoren zu und von Ihren Apache-Kafka-Clustern.

Bewährte Methoden

Verwenden Sie diese Informationen, um schnell Empfehlungen zur Maximierung der Leistung mit Amazon Connect zu finden.

Themen

- Verbindung über Konnektoren herstellen

Verbindung über Konnektoren herstellen

Die folgenden bewährten Methoden können die Leistung Ihrer Konnektivität mit Amazon MSK Connect verbessern.

Die IP-Adressen für Amazon VPC Peering oder Transit Gateway dürfen sich nicht überschneiden

Wenn Sie Amazon-VPC-Peering oder Transit Gateway mit Amazon MSK Connect verwenden, konfigurieren Sie den Konnektor nicht für den Zugriff auf die gepeerten VPC-Ressourcen mit IP-Adressen in den CIDR-Bereichen:

- „10.99.0.0/16“
- „192.168.0.0/16“
- „172.21.0.0/16“

Leitfaden zur Amazon MSK Connect-Migration

In diesem Abschnitt wird beschrieben, wie Sie Ihre Apache Kafka Connector-Anwendung zu Amazon Managed Streaming for Apache Kafka Connect (Amazon MSK Connect) migrieren.

Themen

Vorteile der Verwendung von Amazon MSK Connect

Apache Kafka ist eine der am weitesten verbreiteten Open-Source-Streaming-Plattformen für die Aufnahme und Verarbeitung von Echtzeit-Datenströmen. Mit Apache Kafka können Sie Ihre datenproduzierenden und datenverbrauchenden Anwendungen entkoppeln und unabhängig voneinander skalieren.

Kafka Connect ist eine wichtige Komponente beim Erstellen und Ausführen von Streaming-Anwendungen mit Apache Kafka. Kafka Connect bietet eine standardisierte Methode zum Verschieben von Daten zwischen Kafka und externen Systemen. Kafka Connect ist hochgradig skalierbar und kann große Datenmengen verarbeiten. Kafka Connect bietet leistungsstarke API-Operationen und Tools für die Konfiguration, Bereitstellung und Überwachung von Konnektoren, die Daten zwischen Kafka-Themen und externen Systemen übertragen. Sie können diese Tools verwenden, um die Funktionalität von Kafka Connect an die spezifischen Anforderungen Ihrer Streaming-Anwendung anzupassen und zu erweitern.

Sie können auf Probleme stoßen, wenn Sie Apache Kafka Connect-Cluster eigenständig betreiben oder wenn Sie versuchen, Open-Source-Apache Kafka Connect-Anwendungen zu migrieren. AWS Zu diesen Herausforderungen gehören der Zeitaufwand für die Einrichtung der Infrastruktur und die Bereitstellung von Anwendungen, technische Hindernisse bei der Einrichtung von selbstverwalteten Apache Kafka Connect-Clustern und der administrative Betriebsaufwand.

Um diesen Herausforderungen zu begegnen, empfehlen wir, Amazon Managed Streaming for Apache Kafka Connect (Amazon MSK Connect) zu verwenden, um Ihre Open-Source-Apache Kafka Connect-Anwendungen zu migrieren. AWS Amazon MSK Connect vereinfacht die Verwendung von Kafka Connect zum Streamen von und zu Apache Kafka-Clustern und externen Systemen wie Datenbanken, Suchindizes und Dateisystemen.

Hier sind einige der Vorteile einer Migration zu Amazon MSK Connect:

- Eliminierung des betrieblichen Overheads — Amazon MSK Connect verringert den betrieblichen Aufwand, der mit dem Patchen, Bereitstellen und Skalieren von Apache Kafka Connect-Clustern verbunden ist. Amazon MSK Connect überwacht kontinuierlich den Zustand Ihrer Connect-Cluster und automatisiert Patches und Versions-Upgrades, ohne Ihre Workloads zu stören.
- Automatischer Neustart von Connect-Aufgaben — Amazon MSK Connect kann fehlgeschlagene Aufgaben automatisch wiederherstellen, um Produktionsunterbrechungen zu reduzieren. Aufgabenausfälle können durch vorübergehende Fehler verursacht werden, z. B. durch das Überschreiten des TCP-Verbindungslimits für Kafka und durch eine Neuverteilung von Aufgaben, wenn neue Mitarbeiter der Nutzergruppe für Senk-Connectoren beitreten.
- Automatische horizontale und vertikale Skalierung — Amazon MSK Connect ermöglicht es der Connector-Anwendung, automatisch zu skalieren, um höhere Durchsätze zu unterstützen. Amazon MSK Connect verwaltet die Skalierung für Sie. Sie müssen nur die Anzahl der Mitarbeiter in der Auto Scaling-Gruppe und die Nutzungsschwellenwerte angeben. Sie können den Amazon MSK Connect `UpdateConnector` API-Vorgang verwenden, um die vCPUs zwischen 1 und 8 vCPUs vertikal nach oben oder unten zu skalieren, um einen variablen Durchsatz zu unterstützen.
- Private Netzwerkkonnektivität — Amazon MSK Connect stellt über private DNS-Namen eine private Verbindung zu Quell AWS PrivateLink - und Senkensystemen her.

Migration zu Amazon MSK Connect

In diesem Abschnitt werden kurz die von Kafka Connect und Amazon MSK Connect verwendeten Themen zur Statusverwaltung beschrieben. In diesem Abschnitt werden auch Verfahren für die Migration von Quell- und Senken-Konnektoren behandelt.

Themen

Interne Themen, die von Kafka Connect verwendet werden

Eine Apache Kafka Connect-Anwendung, die im verteilten Modus ausgeführt wird, speichert ihren Status mithilfe interner Themen im Kafka-Cluster und der Gruppenmitgliedschaft. Die folgenden Konfigurationswerte entsprechen den internen Themen, die für Kafka Connect-Anwendungen verwendet werden:

- Thema zur Konfiguration, spezifiziert durch `config.storage.topic`

  Im Thema Konfiguration speichert Kafka Connect die Konfiguration aller Konnektoren und Aufgaben, die von Benutzern gestartet wurden. Jedes Mal, wenn Benutzer die Konfiguration eines Connectors aktualisieren oder wenn ein Connector eine Neukonfiguration anfordert (wenn der Connector beispielsweise feststellt, dass er weitere Aufgaben starten kann), wird ein Datensatz zu diesem Thema ausgegeben. Für dieses Thema ist die Komprimierung aktiviert, sodass immer der letzte Status für jede Entität beibehalten wird.
- Thema Offsets, spezifiziert durch `offset.storage.topic`

  Im Thema Offsets speichert Kafka Connect die Offsets der Quellkonnektoren. Wie beim Thema Konfiguration geht es auch beim Thema Offsets darum, dass die Komprimierung aktiviert ist. Dieses Thema wird nur zum Schreiben der Quellpositionen für Quellkonnektoren verwendet, die Daten aus externen Systemen für Kafka erzeugen. Sink-Konnektoren, die Daten von Kafka lesen und an externe Systeme senden, speichern ihre Verbraucher-Offsets mithilfe regulärer Kafka-Verbrauchergruppen.
- Statusthema, spezifiziert durch `status.storage.topic`

  Im Statusthema speichert Kafka Connect den aktuellen Status von Konnektoren und Aufgaben. Dieses Thema wird als zentraler Ort für die Daten verwendet, die von Benutzern der REST-API abgefragt werden. Dieses Thema ermöglicht es Benutzern, jeden beliebigen Worker abzufragen und trotzdem den Status aller laufenden Plugins abzurufen. Wie bei den Themen Konfiguration und Offsets ist auch beim Thema Status die Komprimierung aktiviert.

Zusätzlich zu diesen Themen nutzt Kafka Connect in großem Umfang die Gruppenmitgliedschafts-API von Kafka. Die Gruppen sind nach dem Namen des Connectors benannt. Bei einem Connector mit dem Namen file-sink wird die Gruppe beispielsweise benannt. connect-file-sink Jeder Benutzer in der Gruppe stellt Datensätze für eine einzelne Aufgabe bereit. Diese Gruppen und ihre Offsets können mithilfe herkömmlicher Tools für Nutzergruppen abgerufen werden, z. B. `Kafka-consumer-group.sh` Für jeden Sink-Connector führt die Connect-Laufzeit eine reguläre Consumer-Gruppe aus, die Datensätze aus Kafka extrahiert.

Statusverwaltung von Amazon MSK Connect-Anwendungen

Standardmäßig erstellt Amazon MSK Connect drei separate Themen im Kafka-Cluster für jeden Amazon MSK Connector, um die Konfiguration, den Offset und den Status des Connectors zu speichern. Die Standard-Themennamen sind wie folgt strukturiert:

- *__msk_connect_configs_Konnektorname _ Konnektor-ID*
- *__msk_connect_status_ Verbindungsname _ Konnektor-ID*
- *__msk_connect_offsets_ Verbindungsname _ Konnektor-ID*

> **Note**
>
> Um die Offset-Kontinuität zwischen den Quell-Connectoren zu gewährleisten, können Sie anstelle des Standardthemas ein Offset-Speicher-Thema Ihrer Wahl verwenden. Wenn Sie ein Offset-Speicherthema angeben, können Sie Aufgaben wie das Erstellen eines Quell-Konnektors erledigen, der den Lesevorgang vom letzten Offset eines vorherigen Konnektors aus wieder aufnimmt. Um ein Offset-Storage-Thema anzugeben, geben Sie einen Wert für die offset.storage.topicEigenschaft in der Amazon MSK Connect-Worker-Konfiguration ein, bevor Sie den Connector erstellen.

Migration von Quell-Connectoren zu Amazon MSK Connect

Source Connectors sind Apache Kafka Connect-Anwendungen, die Datensätze aus externen Systemen in Kafka importieren. In diesem Abschnitt wird der Prozess für die Migration von Apache Kafka Connect Source Connect-Anwendungen beschrieben, die lokale oder selbstverwaltete Kafka Connect-Cluster ausführen, die auf AWS Amazon MSK Connect ausgeführt werden.

Die Anwendung Kafka Connect Source Connector speichert Offsets in einem Thema, das mit dem Wert benannt ist, der für die Eigenschaft config festgelegt ist. `offset.storage.topic` Im

Folgenden finden Sie Beispiele für Offsetnachrichten für einen JDBC-Connector, der zwei Aufgaben ausführt, die Daten aus zwei verschiedenen Tabellen mit dem Namen und importieren. `movies` `shows` Die zuletzt aus der Tabelle movies importierte Zeile hat die primäre ID. `18343` Die zuletzt aus der Tabelle Shows importierte Zeile hat die primäre ID`732`.

```
["jdbcsource",{"protocol":"1","table":"sample.movies"}] {"incrementing":18343}
["jdbcsource",{"protocol":"1","table":"sample.shows"}] {"incrementing":732}
```

Gehen Sie wie folgt vor, um Quell-Connectors zu Amazon MSK Connect zu migrieren:

1. Erstellen Sie ein benutzerdefiniertes Amazon MSK Connect-Plug-in, indem Sie Connector-Bibliotheken aus Ihrem lokalen oder selbstverwalteten Kafka Connect-Cluster abrufen.
2. Erstellen Sie Amazon MSK Connect-Worker-Eigenschaften und legen Sie die Eigenschaften `key.converter`value.converter, und `offset.storage.topic` auf dieselben Werte fest, die für den Kafka-Konnektor festgelegt sind, der in Ihrem vorhandenen Kafka Connect-Cluster ausgeführt wird.
3. Halten Sie die Connector-Anwendung auf dem vorhandenen Cluster an, indem `PUT /connectors/`*`connector-name`*`/pause` Sie eine Anfrage auf dem vorhandenen Kafka Connect-Cluster stellen.
4. Stellen Sie sicher, dass alle Aufgaben der Connector-Anwendung vollständig beendet sind. Sie können die Aufgaben beenden, indem Sie entweder eine `GET /connectors/`*`connector-name`*`/status` Anfrage im vorhandenen Kafka Connect-Cluster stellen oder indem Sie die Nachrichten aus dem Themennamen verwenden, der für die Eigenschaft `status.storage.topic` festgelegt ist.
5. Rufen Sie die Konnektorkonfiguration aus dem vorhandenen Cluster ab. Sie können die Konnektorkonfiguration abrufen, indem Sie entweder eine `GET /connectors/`*`connector-name`*`/config/` Anfrage für den vorhandenen Cluster stellen oder indem Sie die Nachrichten aus dem Themennamen verwenden, der für die Eigenschaft festgelegt ist`config.storage.topic`.
6. Erstellen Sie einen neuen Amazon MSK Connector mit demselben Namen wie ein vorhandener Cluster. Erstellen Sie diesen Connector mithilfe des benutzerdefinierten Connector-Plug-ins, das Sie in Schritt 1 erstellt haben, der Worker-Eigenschaften, die Sie in Schritt 2 erstellt haben, und der Connector-Konfiguration, die Sie in Schritt 5 extrahiert haben.
7. Wenn der Amazon MSK Connector-Status lautet`active`, überprüfen Sie anhand der Protokolle, ob der Connector mit dem Import von Daten aus dem Quellsystem begonnen hat.
8. Löschen Sie den Connector im vorhandenen Cluster, indem `DELETE /connectors/`*`connector-name`* Sie eine Anfrage stellen.

Migration von Sink-Konnektoren zu Amazon MSK Connect

Sink Connectors sind Apache Kafka Connect-Anwendungen, die Daten von Kafka in externe Systeme exportieren. In diesem Abschnitt wird der Prozess für die Migration von Apache Kafka Connect Sink Connector-Anwendungen beschrieben, auf denen lokale oder selbstverwaltete Kafka Connect-Cluster ausgeführt werden, die auf AWS Amazon MSK Connect ausgeführt werden.

Kafka Connect-Sink-Konnektoren verwenden die Kafka-API für Gruppenmitgliedschaft und speichern Offsets in denselben `__consumer_offset` Themen wie eine typische Verbraucheranwendung. Dieses Verhalten vereinfacht die Migration des Sink-Connectors von einem selbstverwalteten Cluster zu Amazon MSK Connect.

Gehen Sie wie folgt vor, um Sink Connectors zu Amazon MSK Connect zu migrieren:

1. Erstellen Sie ein benutzerdefiniertes Amazon MSK Connect-Plug-in, indem Sie Connector-Bibliotheken aus Ihrem lokalen oder selbstverwalteten Kafka Connect-Cluster abrufen.
2. Erstellen Sie Amazon MSK Connect-Worker-Eigenschaften und legen Sie die Eigenschaften `key.converter` und `value.converter` auf dieselben Werte fest, die für den Kafka-Konnektor festgelegt sind, der in Ihrem vorhandenen Kafka Connect-Cluster ausgeführt wird.
3. Halten Sie die Connector-Anwendung auf Ihrem vorhandenen Cluster an, indem Sie eine `PUT /connectors/`*`connector-name`*`/pause` Anfrage auf dem vorhandenen Kafka Connect-Cluster stellen.
4. Stellen Sie sicher, dass alle Aufgaben der Connector-Anwendung vollständig beendet sind. Sie können die Aufgaben beenden, indem Sie entweder eine `GET /connectors/`*`connector-name`*`/status` Anfrage auf dem vorhandenen Kafka Connect-Cluster stellen oder indem Sie die Nachrichten aus dem Themennamen verwenden, der für die Eigenschaft `status.storage.topic` festgelegt ist.
5. Rufen Sie die Konnektorkonfiguration aus dem vorhandenen Cluster ab. Sie können die Konnektorkonfiguration entweder abrufen, indem Sie eine `GET /connectors/`*`connector-name`*`/config` Anfrage für den vorhandenen Cluster stellen oder indem Sie die Nachrichten aus dem Themennamen verwenden, der für die Eigenschaft festgelegt ist`config.storage.topic`.
6. Erstellen Sie einen neuen Amazon MSK Connector mit demselben Namen wie der bestehende Cluster. Erstellen Sie diesen Connector mithilfe des benutzerdefinierten Connector-Plug-ins, das Sie in Schritt 1 erstellt haben, der Worker-Eigenschaften, die Sie in Schritt 2 erstellt haben, und der Connector-Konfiguration, die Sie in Schritt 5 extrahiert haben.
7. Wenn der Amazon MSK Connector-Status lautet`active`, überprüfen Sie anhand der Protokolle, ob der Connector mit dem Import von Daten aus dem Quellsystem begonnen hat.

8. Löschen Sie den Connector im vorhandenen Cluster, indem `DELETE /connectors/connector-name` Sie eine Anfrage stellen.

Fehlerbehebung bei Amazon MSK Connect

Die folgenden Informationen können zum Beheben von Problemen nützlich sein, die Sie bei der Verwendung von MSK Connect haben könnten. Sie können Ihr Problem auch im AWS re:Post posten.

Der Konnektor kann nicht auf Ressourcen zugreifen, die im öffentlichen Internet gehostet werden

Siehe Aktivieren des Internetzugangs für Amazon MSK Connect.

Die Anzahl der laufenden Aufgaben im Konnektor entspricht nicht der Anzahl der in tasks.max angegebenen Aufgaben

Hier sind einige Gründe, warum ein Konnektor möglicherweise weniger Aufgaben als die angegebene tasks.max-Konfiguration verwendet:

- Einige Konnektor-Implementierungen begrenzen die Anzahl der Aufgaben, die verwendet werden können. Zum Beispiel ist der Debezium-Konnektor für MySQL auf die Verwendung einer einzigen Aufgabe beschränkt.
- Bei Verwendung des automatisch skalierten Kapazitätsmodus überschreibt Amazon MSK Connect die tasks.max-Eigenschaft eines Konnektors mit einem Wert, der proportional zur Anzahl der Worker, die im Konnektor laufen, und zur Anzahl der MCUs pro Worker ist.
- Bei Sink-Konnektoren darf der Grad der Parallelität (Anzahl der Aufgaben) nicht höher sein als die Anzahl der Themenpartitionen. Sie können den Wert tasks.max zwar größer einstellen, aber eine einzelne Partition wird nie von mehr als einer einzelnen Aufgabe gleichzeitig verarbeitet.
- In Kafka Connect 2.7.x ist der standardmäßige Verbraucher-Partitionszuweiser `RangeAssignor`. Das Verhalten dieses Zuweisers besteht darin, die erste Partition jedes Themas einem einzelnen Verbraucher zuzuweisen, die zweite Partition jedes Themas einem einzelnen Verbraucher usw. Das bedeutet, dass die maximale Anzahl von aktiven Aufgaben für einen Sink-Konnektor, der `RangeAssignor` verwendet, der maximalen Anzahl von Partitionen in einem einzelnen Thema entspricht, die verwendet werden. Wenn dies für Ihren Anwendungsfall nicht funktioniert, sollten Sie eine Worker-Konfiguration erstellen, in der die Eigenschaft `consumer.partition.assignment.strategy` auf einen geeigneteren Verbraucher-Partitionszuweiser gesetzt ist. Siehe Kafka 2.7-Schnittstelle ConsumerPartitionAssignor: Alle bekannten Implementierungsklassen.

MSK-Replikator

Was ist Amazon MSK Replicator?

Amazon MSK Replicator ist eine Amazon MSK-Funktion, mit der Sie Daten zuverlässig über Amazon MSK-Cluster in verschiedenen oder derselben AWS Region (en) replizieren können. Mit MSK-Replikator können Sie auf einfache Weise regional belastbare Streaming-Anwendungen erstellen, um die Verfügbarkeit und Geschäftskontinuität zu erhöhen. MSK-Replikator bietet automatische asynchrone Replikation über MSK-Cluster hinweg, sodass Sie keinen benutzerdefinierten Code schreiben, die Infrastruktur verwalten oder regionsübergreifende Netzwerke einrichten müssen.

MSK-Replikator skaliert automatisch die zugrunde liegenden Ressourcen, sodass Sie Daten bei Bedarf replizieren können, ohne die Kapazität überwachen oder skalieren zu müssen. MSK-Replikator repliziert auch die erforderlichen Kafka-Metadaten, einschließlich Themenkonfigurationen, Zugriffssteuerungslisten (ACLs) und Verbrauchergruppen-Offsets. Wenn in einer Region ein unerwartetes Ereignis eintritt, können Sie ein Failover auf die andere AWS Region durchführen und die Verarbeitung nahtlos fortsetzen.

MSK-Replikator unterstützt sowohl die regionsübergreifende Replikation (CRR) als auch die regionsinterne Replikation (SRR). Bei der regionsübergreifenden Replikation befinden sich die MSK-Quell- und Zielcluster in unterschiedlichen Regionen. AWS Bei der Replikation in derselben Region befinden sich sowohl der Quell- als auch der Ziel-MSK-Cluster in derselben Region. AWS Sie müssen Quell- und Ziel-MSK-Cluster erstellen, bevor Sie sie mit MSK-Replikator verwenden können.

> **Note**
>
> MSK Replicator unterstützt die folgenden AWS Regionen: USA Ost (us-east-1, Nord-Virginia); USA Ost (us-east-2, Ohio); USA West (us-west-2, Oregon); Europa (eu-west-1, Irland); Europa (eu-central-1, Frankfurt); Asien-Pazifik (ap-southeast-1, Singapur); Asien-Pazifik (ap-southeast-2, Sydney), Europa (eu-north-1, Stockholm), Asien-Pazifik (ap-south-1, Mumbai), Europa (eu-west-3, Paris), Südamerika (sa-east-1, São Paulo), Asien-Pazifik (ap-northeast-2, Seoul), Europa (eu-west-2, London), Asien-Pazifik (ap-northeast-1, Tokio), USA West (us-west-1, Nordkalifornien), Kanada (ca-central-1, zentral).

Im Folgenden finden Sie einige häufig verwendete Anwendungen für Amazon MSK Replicator.

- Streaming-Anwendungen für mehrere Regionen erstellen: Erstellen Sie hochverfügbare und fehlertolerante Streaming-Anwendungen für mehr Stabilität, ohne benutzerdefinierte Lösungen einrichten zu müssen.
- Datenzugriff mit niedrigerer Latenz: Bieten Sie Verbrauchern in verschiedenen geografischen Regionen Datenzugriff mit niedrigerer Latenz.
- Daten an Ihre Partner verteilen: Kopieren Sie Daten von einem Apache-Kafka-Cluster in viele Apache-Kafka-Cluster, sodass verschiedene Teams/Partner ihre eigenen Datenkopien haben.
- Daten für Analysen aggregieren: Kopieren Sie Daten aus mehreren Apache-Kafka-Clustern in einen Cluster, um auf einfache Weise Einblicke in aggregierte Echtzeitdaten zu gewinnen.
- Lokal schreiben, global auf Ihre Daten zugreifen: Richten Sie die multiaktive Replikation ein, um in einer AWS Region durchgeführte Schreibvorgänge automatisch auf andere Regionen zu übertragen, um Daten mit geringerer Latenz und geringeren Kosten bereitzustellen.

Funktionsweise von Amazon MSK Replicator

Um mit MSK Replicator zu beginnen, müssen Sie einen neuen Replicator in der Region Ihres Zielclusters erstellen. AWS MSK Replicator kopiert automatisch alle Daten aus dem Cluster in der primären AWS Region, die als Quelle bezeichnet wird, in den Cluster in der Zielregion, der Zielregion genannt wird. Quell- und Zielcluster können sich in derselben oder in unterschiedlichen AWS Regionen befinden. Sie müssen den Ziel-Cluster erstellen, wenn er nicht bereits vorhanden ist.

Wenn Sie einen Replikator erstellen, stellt MSK Replicator alle erforderlichen Ressourcen in der AWS Region des Zielclusters bereit, um die Latenz bei der Datenreplikation zu optimieren. Die Replikationslatenz hängt von vielen Faktoren ab, darunter der Netzwerkentfernung zwischen den AWS Regionen Ihrer MSK-Cluster, der Durchsatzkapazität Ihrer Quell- und Zielcluster und der Anzahl der Partitionen auf Ihren Quell- und Zielclustern. MSK-Replikator skaliert automatisch die zugrunde liegenden Ressourcen, sodass Sie Daten bei Bedarf replizieren können, ohne die Kapazität überwachen oder skalieren zu müssen.

Datenreplikation

Standardmäßig kopiert MSK Replicator alle Daten asynchron vom letzten Offset in den Themenpartitionen des Quellclusters in den Zielcluster. Wenn die Einstellung „Neue Themen erkennen und kopieren“ aktiviert ist, erkennt MSK Replicator automatisch neue Themen oder Themenpartitionen und kopiert sie in den Zielcluster. Es kann jedoch bis zu 30 Sekunden dauern, bis der Replicator die neuen Themen oder Themenpartitionen auf dem Zielcluster erkennt und erstellt.

Alle Nachrichten, die an das Quellthema gesendet wurden, bevor das Thema auf dem Zielcluster erstellt wurde, werden nicht repliziert. Alternativ können Sie Ihren Replicator bei der Erstellung so konfigurieren, dass er die Replikation ab dem frühesten Offset in den Themenpartitionen des Quell-Clusters startet, wenn Sie vorhandene Nachrichten zu Ihren Themen auf den Zielcluster replizieren möchten.

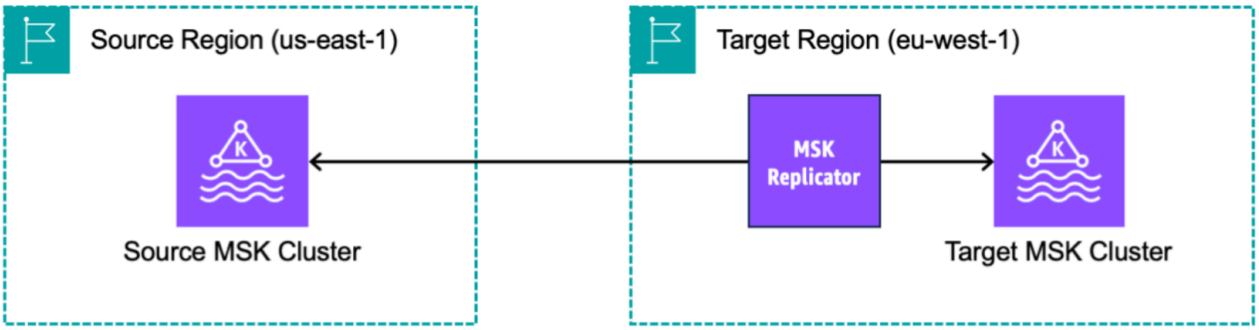
Der MSK Replicator speichert Ihre Daten nicht. Daten werden aus Ihrem Quellcluster abgerufen, im Arbeitsspeicher gepuffert und in den Zielcluster geschrieben. Der Puffer wird automatisch gelöscht, wenn die Daten entweder erfolgreich geschrieben wurden oder nach erneuten Versuchen fehlschlagen. Die gesamte Kommunikation und die Daten zwischen MSK Replicator und Ihren Clustern werden bei der Übertragung immer verschlüsselt. Alle MSK Replicator API-Aufrufe wie`DescribeClusterV2`,, `CreateTopic` werden in erfasst. `DescribeTopicDynamicConfiguration` AWS CloudTrail Ihre MSK-Broker-Protokolle werden dasselbe wiedergeben.

MSK Replicator erstellt Themen im Zielcluster mit einem Replicator-Faktor von 3. Bei Bedarf können Sie den Replikationsfaktor direkt auf dem Zielcluster ändern.

Replikation von Metadaten

MSK Replicator unterstützt auch das Kopieren der Metadaten vom Quellcluster in den Zielcluster. Zu den Metadaten gehören die Themenkonfiguration, Lesezugriffskontrolllisten (ACLs) und Offsets für Nutzergruppen. Wie die Datenreplikation erfolgt auch die Metadatenreplikation asynchron. Um eine bessere Leistung zu erzielen, priorisiert MSK Replicator die Datenreplikation gegenüber der Metadatenreplikation.

Als Teil der Offset-Synchronisierung für Nutzergruppen optimiert MSK Replicator die Daten für Ihre Benutzer auf dem Quell-Cluster, die von einer Position aus lesen, die näher an der Spitze des Streams liegt (Ende der Themenpartition). Wenn Ihre Nutzergruppen im Quell-Cluster hinterherhinken, können Sie bei diesen Nutzergruppen auf dem Ziel-Cluster eine höhere Verzögerung feststellen als beim Quell-Cluster. Das bedeutet, dass Ihre Kunden nach einem Failover auf den Zielcluster mehr doppelte Nachrichten erneut verarbeiten werden. Um diese Verzögerung zu verringern, müssten Ihre Verbraucher auf dem Quell-Cluster aufholen und von der Spitze des Streams (Ende der Themenpartition) aus mit dem Konsum beginnen. Wenn Ihre Kunden aufholen, reduziert MSK Replicator die Verzögerung automatisch.



Anforderungen und Überlegungen zum Erstellen eines Amazon MSK Replicators

Beachten Sie diese MSK-Cluster-Anforderungen für den Betrieb eines Amazon MSK Replicators.

Themen

- Gewährt die Berechtigung zum Erstellen eines MSK-Replikators
- Unterstützte Clustertypen und Versionen
- MSK-Serverless-Cluster-Konfiguration
- Änderungen der Cluster-Konfiguration

Gewährt die Berechtigung zum Erstellen eines MSK-Replikators

Hier ist ein Beispiel für die IAM-Richtlinie, die für die Erstellung eines MSK-Replikators erforderlich ist. Die Aktion `kafka:TagResource` ist nur erforderlich, wenn bei der Erstellung des MSK-Replikators Tags angegeben werden.

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

```
                "iam:PassRole",
                "iam:CreateServiceLinkedRole",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:CreateNetworkInterface",
                "ec2:DescribeVpcs",
                "kafka:CreateReplicator",
                "kafka:TagResource"
            ],
            "Resource": "*"
        }
    ]
}
```

Es folgt ein Beispiel einer IAM-Richtlinie zur Beschreibung des Replikators. Entweder die Aktion `kafka:DescribeReplicator` oder die Aktion `kafka:ListTagsForResource` ist erforderlich, nicht beides.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "kafka:DescribeReplicator",
                "kafka:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

Unterstützte Clustertypen und Versionen

Dies sind Anforderungen für unterstützte Instance-Typen, Kafka-Versionen und Netzwerkkonfigurationen.

- MSK-Replikator unterstützt sowohl von MSK bereitgestellte Cluster als auch MSK-Serverless-Cluster in beliebiger Kombination als Quell- und Ziel-Cluster. Andere Arten von Kafka-Clustern werden derzeit von MSK-Replikator nicht unterstützt.

- Serverless-MSK-Cluster erfordern eine IAM-Zugriffssteuerung, unterstützen keine Apache-Kafka-ACL-Replikation und die themenspezifische Konfigurationsreplikation wird nur eingeschränkt unterstützt. Siehe MSK Serverless.
- MSK Replicator wird nur auf Clustern unterstützt, auf denen Apache Kafka 2.7.0 oder höher ausgeführt wird, unabhängig davon, ob sich Ihre Quell- und Zielcluster in derselben oder in unterschiedlichen Regionen befinden. AWS
- MSK-Replikator unterstützt Cluster, die Instance-Typen m5.large oder größer verwenden. t3.small-Cluster werden nicht unterstützt.
- Wenn Sie MSK-Replikator mit einem von MSK bereitgestellten Cluster verwenden, benötigen Sie mindestens je drei Broker in Quell- und Ziel-Clustern. Sie können Daten clusterübergreifend in zwei Availability Zones replizieren, benötigen jedoch mindestens vier Broker in diesen Clustern.
- Sowohl Ihr Quell- als auch Ihr Ziel-MSK-Cluster müssen sich im selben Konto befinden. AWS Die Replikation zwischen Clustern in verschiedenen Konten wird nicht unterstützt.
- Wenn sich die Quell- und Ziel-MSK-Cluster in unterschiedlichen AWS Regionen (regionsübergreifend) befinden, verlangt MSK Replicator, dass für den Quellcluster private Multi-VPC-Konnektivität für seine IAM-Zugriffskontrollmethode aktiviert ist. Multi-VPC ist für andere Authentifizierungsmethoden auf dem Quell-Cluster nicht erforderlich. Multi-VPC ist nicht erforderlich, wenn Sie Daten zwischen Clustern in derselben Region replizieren. AWS Siehe the section called “Private Multi-VPC-Konnektivität in einer einzelnen Region”.

MSK-Serverless-Cluster-Konfiguration

- MSK Serverless unterstützt die Replikation dieser Themenkonfigurationen für MSK-Serverless-Ziel-Cluster während der Themenerstellung: `cleanup.policy`, `compression.type`, `max.message.bytes`, `retention.bytes`, `retention.ms`.
- MSK Serverless unterstützt während der Synchronisierung der Themenkonfiguration nur diese Themenkonfigurationen: `compression.type`, `max.message.bytes`, `retention.bytes`, `retention.ms`.
- Replikator verwendet 83 komprimierte Partitionen auf MSK-Serverless-Ziel-Clustern. Stellen Sie sicher, dass die MSK-Serverless-Ziel-Cluster über eine ausreichende Anzahl komprimierter Partitionen verfügen. Siehe MSK-Serverless-Kontingent.

Änderungen der Cluster-Konfiguration

- Es wird empfohlen, den gestaffelten Speicher nicht ein- oder auszuschalten, nachdem der MSK-Replikator erstellt wurde. Wenn Ihr Ziel-Cluster nicht mehrstufig ist, kopiert MSK die gestaffelte Speicherkonfigurationen nicht, unabhängig davon, ob Ihr Quell-Cluster gestaffelt ist oder nicht. Wenn Sie nach der Erstellung des Replikators den gestaffelten Speicher auf dem Ziel-Cluster aktivieren, muss der Replikator neu erstellt werden. Wenn Sie Daten von einem nicht-mehrstufigen Cluster in einen mehrstufigen Cluster kopieren möchten, sollten Sie keine Themenkonfigurationen kopieren. Weitere Informationen finden Sie unter Aktivieren und Deaktivieren der gestaffelten Speicherung bei einem vorhandenen Thema.
- Ändern Sie die Cluster-Konfigurationseinstellungen nach der Erstellung des MSK-Replikators nicht. Die Cluster-Konfigurationseinstellungen werden bei der Erstellung des MSK-Replikators überprüft. Um Probleme mit dem MSK-Replikator zu vermeiden, sollten Sie die folgenden Einstellungen nicht ändern, nachdem der MSK-Replikator erstellt wurde.
  - Den MSK-Cluster in den Instance-Typ t3 ändern.
  - Berechtigungen für die Service-Ausführungsrolle ändern.
  - Die private MSK-Multi-VPC-Konnektivität deaktivieren.
  - Die angefügte ressourcenbasierte Richtlinie für den Cluster ändern.
  - Die Regeln der Cluster-Sicherheitsgruppe ändern.

Erste Schritte mit Amazon MSK Replicator

Dieses Tutorial zeigt Ihnen, wie Sie einen Quell-Cluster und einen Ziel-Cluster in derselben AWS Region oder in verschiedenen Regionen einrichten. AWS Anschließend verwenden Sie diese Cluster, um einen Amazon MSK Replicator zu erstellen.

Schritt 1: Den Amazon-MSK-Quell-Cluster vorbereiten

Wenn Sie bereits einen MSK-Quell-Cluster für den MSK-Replikator erstellt haben, stellen Sie sicher, dass er die in diesem Abschnitt beschriebenen Anforderungen erfüllt. Gehen Sie andernfalls wie folgt vor, um einen von MSK bereitgestellten Cluster oder einen Serverless-Quell-Cluster zu erstellen.

Das Verfahren zum Erstellen eines regionsübergreifenden und regionsinternen MSK-Replikator-Quell-Clusters ist ähnlich. Unterschiede werden in den folgenden Verfahren hervorgehoben.

1. Erstellen Sie einen von MSK bereitgestellten Cluster oder einen Serverless-Cluster mit aktivierter IAM-Zugriffssteuerung in der Quellregion. Ihr Quell-Cluster muss über mindestens drei Broker verfügen.
2. Wenn bei einem regionsübergreifenden MSK-Replikator die Quelle ein bereitgestellter Cluster ist, konfigurieren Sie ihn mit aktivierter privater Multi-VPC-Konnektivität für IAM-Zugriffssteuerungs-Schema. Beachten Sie, dass der Authentifizierungstyp „Nicht authentifiziert“ nicht unterstützt wird, wenn Multi-VPC aktiviert ist. Sie müssen die private Multi-VPC-Konnektivität für andere Authentifizierungsschema (mTLS oder SASL/SCRAM) nicht aktivieren. Sie können gleichzeitig mTLS- oder SASL/SCRAM-Authentifizierungsschema für Ihre anderen Clients verwenden, die eine Verbindung zu Ihrem MSK-Cluster herstellen. Sie können private Multi-VPC-Konnektivität in den Cluster-Details der Konsole unter Netzwerkeinstellungen oder mit der `UpdateConnectivity`-API konfigurieren. Siehe Cluster-Besitzer aktiviert Multi-VPC. Wenn es sich bei Ihrem Quell-Cluster um einen Serverless-MSK-Cluster handelt, müssen Sie die private Multi-VPC-Konnektivität nicht aktivieren.

   Für einen regionsinternen MSK-Replikator benötigt der MSK-Quell-Cluster keine private Multi-VPC-Konnektivität, und andere Clients können weiterhin mit dem Authentifizierungstyp „Nicht authentifiziert“ auf den Cluster zugreifen.
3. Für regionsübergreifende MSK-Replikatoren müssen Sie dem Quell-Cluster eine ressourcenbasierte Berechtigungsrichtlinie hinzufügen. Dadurch kann MSK eine Verbindung zu diesem Cluster herstellen, um Daten zu replizieren. Sie können dies mithilfe der folgenden CLI- oder AWS Konsolenverfahren tun. Siehe auch ressourcenbasierte Amazon-MSK-Richtlinien. Dieser Schritt ist für regionsinterne MSK-Replikatoren nicht nötig.

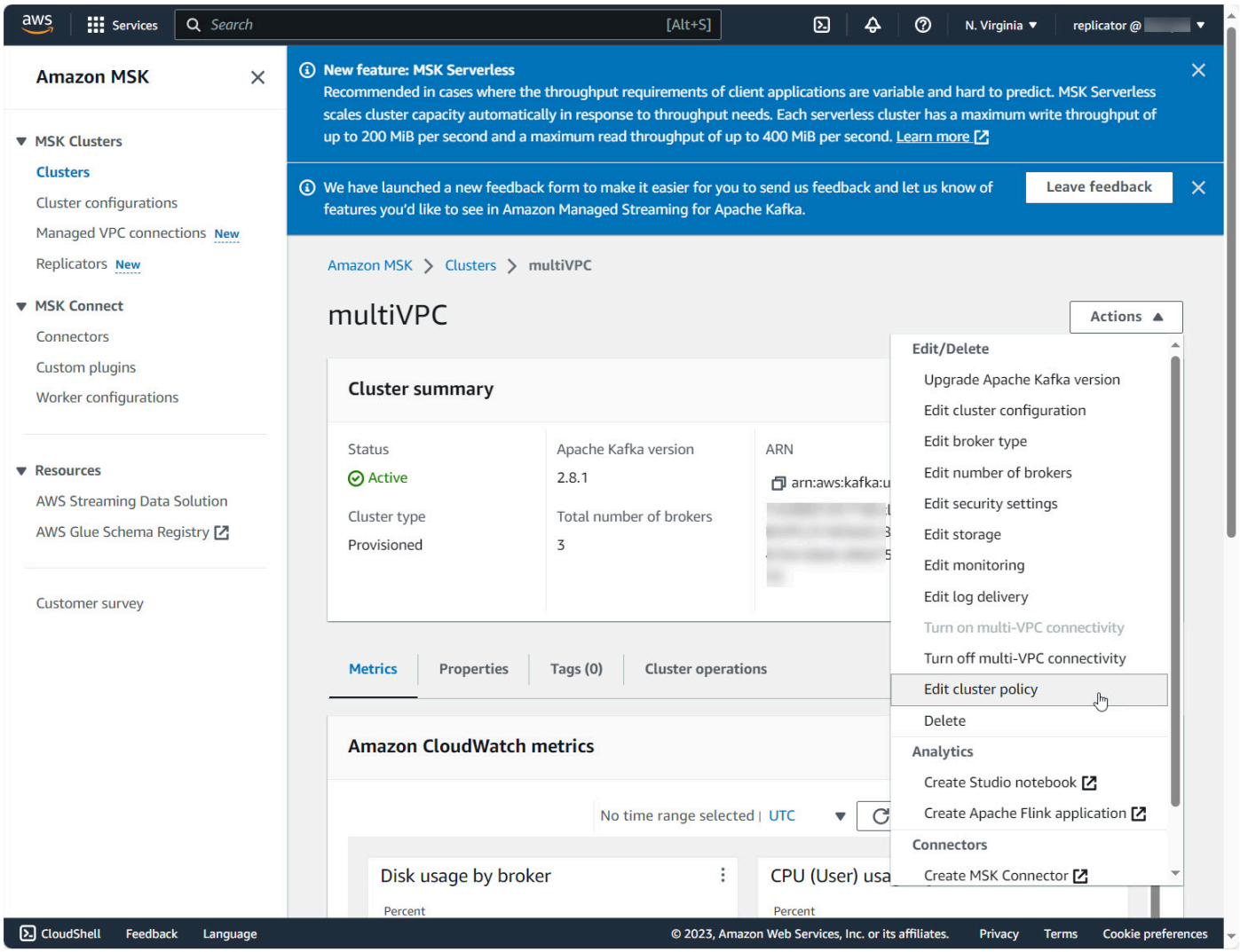
Console: create resource policy

Aktualisieren Sie die Quell-Cluster-Richtlinie mit dem folgenden JSON-Code. Ersetzen Sie den Platzhalter durch den ARN Ihres Quell-Clusters.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Service": [
                "kafka.amazonaws.com"
            ]
```

```
        },
        "Action": [
            "kafka:CreateVpcConnection",
            "kafka:GetBootstrapBrokers",
            "kafka:DescribeClusterV2"
        ],
        "Resource": "<sourceClusterARN>"
    }
  ]
}
```

Verwenden Sie die Option Cluster-Richtlinie bearbeiten im Menü Aktionen auf der Cluster-Detailseite.



CLI: create resource policy

Hinweis: Wenn Sie die AWS Konsole verwenden, um einen Quellcluster zu erstellen, und die Option zum Erstellen einer neuen IAM-Rolle wählen, wird die erforderliche Vertrauensrichtlinie an die Rolle AWS angehängt. Wenn MSK hingegen eine vorhandene IAM-Rolle verwenden soll oder wenn Sie selbst eine Rolle erstellen, fügen Sie dieser Rolle die folgende Vertrauensrichtlinie an, damit MSK-Replikator sie annehmen kann. Weitere Informationen zum Ändern der Vertrauensstellung einer Rolle finden Sie unter Ändern einer Rolle.

1. Rufen Sie mit diesem Befehl die aktuelle Version der MSK-Cluster-Richtlinie ab. Ersetzen Sie Platzhalter durch den tatsächlichen Cluster-ARN.

```
aws kafka get-cluster-policy –cluster-arn <Cluster ARN>
{
"CurrentVersion": "K1PA6795UKM GR7",
"Policy": "..."
}
```

2. Erstellen Sie eine ressourcenbasierte Richtlinie, um MSK-Replikator den Zugriff auf den Quell-Cluster zu ermöglichen. Verwenden Sie die folgende Syntax als Vorlage und ersetzen Sie den Platzhalter durch den tatsächlichen Quell-Cluster-ARN.

```
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>"
}
]
```

Schritt 2: Den Amazon-MSK-Ziel-Cluster vorbereiten

Erstellen Sie einen MSK-Ziel-Cluster (bereitgestellt oder serverless) mit aktivierter IAM-Zugriffssteuerung. Für den Ziel-Cluster ist es nicht erforderlich, dass private Multi-VPC-Konnektivität aktiviert ist. Der Zielcluster kann sich in derselben AWS Region oder einer anderen Region wie der Quellcluster befinden. Sowohl der Quell- als auch der Zielcluster müssen sich im selben AWS Konto befinden. Der Quell-Cluster muss über mindestens drei Broker verfügen.

Schritt 3: Einen Amazon MSK Replicator erstellen

Bevor Sie den Amazon MSK Replicator erstellen, stellen Sie sicher, dass Sie Gewährt die Berechtigung zum Erstellen eines MSK-Replikators haben.

Themen

- Erstellen eines Replikators mithilfe der AWS -Konsole in der Ziel-Cluster-Region
- Wählen Sie den Quell-Cluster
- Wählen Sie den Ziel-Cluster
- Einstellungen und Berechtigungen des Replikators konfigurieren

Erstellen eines Replikators mithilfe der AWS -Konsole in der Ziel-Cluster-Region

1. Öffnen Sie in der AWS Region, in der sich Ihr Ziel-MSK-Cluster befindet, die Amazon MSK-Konsole unter https://console.aws.amazon.com/msk/home?region=us-east-1#/home/.
2. Wählen Sie Replikatoren, um die Liste der Replikatoren im Konto anzuzeigen.
3. Wählen Sie Replikator erstellen.
4. Geben Sie im Replikator-Detailbereich dem neuen Replikator einen eindeutigen Namen.

Wählen Sie den Quell-Cluster

Der Quell-Cluster enthält die Daten, die Sie in einen MSK-Ziel-Cluster kopieren möchten.

1. Wählen Sie im Bereich Quell-Cluster die AWS -Region aus, in der sich der Quell-Cluster befindet.

   Sie können die Region eines Clusters nachschlagen, indem Sie zu MSK-Clusters gehen und sich die Details des Cluster-ARN ansehen. Der Name der Region ist in die ARN-Zeichenfolge eingebettet. Im folgenden Beispiels-ARN ist die Cluster-Region `ap-southeast-2`.

```
arn:aws:kafka:ap-southeast-2:123456789012:cluster/cluster-11/
eec93c7f-4e8b-4baf-89fb-95de01ee639c-s1
```

2. Geben Sie den ARN Ihres Quell-Clusters ein oder suchen Sie nach dem Quell-Cluster, um ihn auszuwählen.
3. Wählen Sie Subnetz(e) für den Quell-Cluster aus.

   In der Konsole werden die Subnetze angezeigt, die in der Region des Quell-Clusters verfügbar sind, sodass Sie sie auswählen können. Sie müssen mindestens zwei Subnetze auswählen. Für einen regionsinternen MSK-Replikator müssen sich die Subnetze, die Sie für den Zugriff auf den Quell-Cluster auswählen, und die Subnetze für den Zugriff auf den Ziel-Cluster in derselben Availability Zone befinden.
4. Wählen Sie Sicherheitsgruppe(n) für den MSK-Replikator zum Zugriff auf den Quell-Cluster aus.

   - Für die regionsübergreifende Replikation (CRR) müssen Sie keine Sicherheitsgruppe (n) für Ihren Quell-Cluster angeben.
   - Rufen Sie für die Replikation derselben Region (SRR) die Amazon EC2 EC2-Konsole unter https://console.aws.amazon.com/ec2/ auf und stellen Sie sicher, dass die Sicherheitsgruppen, die Sie für den Replicator angeben, über Regeln für ausgehenden Datenverkehr zu den Sicherheitsgruppen Ihres Quell-Clusters verfügen. Stellen Sie außerdem sicher, dass die Sicherheitsgruppen Ihres Quell-Clusters über Regeln für eingehenden Datenverkehr von den Replicator-Sicherheitsgruppen verfügen, die für die Quelle bereitgestellt wurden.

     Gehen Sie wie folgt vor, um der Sicherheitsgruppe Ihres Quell-Clusters Regeln für eingehenden Datenverkehr hinzuzufügen:

     1. Gehen Sie in der AWS Konsole zu den Details Ihres Quell-Clusters, indem Sie den Clusternamen auswählen.
     2. Wählen Sie die Registerkarte Eigenschaften und scrollen Sie dann nach unten zum Bereich Netzwerkeinstellungen, um den Namen der angewendeten Sicherheitsgruppe auszuwählen.
     3. Rufen Sie die Regeln für eingehenden Datenverkehr auf und wählen Sie Regeln für eingehenden Datenverkehr bearbeiten.
     4. Wählen Sie Regel hinzufügen.
     5. Wählen Sie in der Spalte Typ für die neue Regel die Option Benutzerdefiniertes TCP aus.

6. Geben Sie in der Spalte Portbereich den Text ein`9098`. MSK Replicator verwendet IAM-Zugriffskontrolle, um eine Verbindung zu Ihrem Cluster herzustellen, der Port 9098 verwendet.
7. Geben Sie in der Spalte Quelle den Namen der Sicherheitsgruppe ein, die Sie bei der Replikatorerstellung für den Quellcluster angeben werden (dies kann mit der Sicherheitsgruppe des MSK-Quellclusters identisch sein), und wählen Sie dann Regeln speichern aus.

Gehen Sie wie folgt vor, um Regeln für ausgehenden Datenverkehr zur Sicherheitsgruppe von Replicator hinzuzufügen, die für die Quelle bereitgestellt wurde:

1. Gehen Sie in der AWS Konsole für Amazon EC2 zu der Sicherheitsgruppe, die Sie bei der Replicator-Erstellung für die Quelle angeben werden.
2. Gehen Sie zu den Regeln für ausgehenden Datenverkehr und wählen Sie Regeln für ausgehenden Datenverkehr bearbeiten aus.
3. Wählen Sie Regel hinzufügen.
4. Wählen Sie in der Spalte Typ für die neue Regel die Option Benutzerdefiniertes TCP aus.
5. Geben Sie in der Spalte Portbereich den Text ein`9098`. MSK Replicator verwendet IAM-Zugriffskontrolle, um eine Verbindung zu Ihrem Cluster herzustellen, der Port 9098 verwendet.
6. Geben Sie in der Spalte Quelle den Namen der Sicherheitsgruppe des MSK-Quellclusters ein, und wählen Sie dann Regeln speichern aus.

> **Note**
>
> Wenn Sie den Datenverkehr nicht mithilfe Ihrer Sicherheitsgruppen einschränken möchten, können Sie alternativ Regeln für eingehenden und ausgehenden Datenverkehr hinzufügen, die den gesamten Datenverkehr zulassen.
> 1. Wählen Sie Regel hinzufügen.
> 2. Wählen Sie in der Spalte Typ die Option Gesamter Datenverkehr aus.
> 3. Geben Sie in der Quelle-Spalte `0.0.0.0/0` ein und wählen Sie dann Regeln speichern.

Wählen Sie den Ziel-Cluster

Der Ziel-Cluster ist der von MSK bereitgestellte Cluster oder der Serverless-Cluster, in den die Quelldaten kopiert werden.

> **Note**
>
> MSK-Replikator erstellt neue Themen im Ziel-Cluster mit einem automatisch generierten Präfix, das dem Themennamen hinzugefügt wird. MSK-Replikator repliziert beispielsweise Daten in „`topic`“ aus dem Quell-Cluster zu einem neuen Thema im Ziel-Cluster namens `<sourceKafkaClusterAlias>.topic`. Dies dient dazu, Themen, die Daten enthalten, die aus dem Quell-Cluster repliziert wurden, von anderen Themen im Ziel-Cluster zu unterscheiden und zu verhindern, dass Daten zwischen den Clustern wiederkehrend repliziert werden. Das Präfix, das den Themennamen im Zielcluster hinzugefügt wird, finden Sie mithilfe der `DescribeReplicator` API im Feld sourceKafkaClusterAlias oder auf der Seite mit den Replicator-Details in der MSK-Konsole. Das Präfix im Zielcluster lautet **<** sourceKafkaCluster Alias**>**.

1. Wählen Sie im Bereich Zielcluster die AWS Region aus, in der sich der Zielcluster befindet.
2. Geben Sie den ARN Ihres Ziel-Clusters ein oder suchen Sie nach dem Ziel-Cluster, um ihn auszuwählen.
3. Wählen Sie Subnetze für den Ziel-Cluster aus.

   In der Konsole werden die Subnetze angezeigt, die in der Region des Ziel-Clusters verfügbar sind, sodass Sie sie auswählen können. Sie müssen mindestens zwei Subnetze auswählen.
4. Wählen Sie Sicherheitsgruppe(n) für den MSK-Replikator zum Zugriff auf den Ziel-Cluster aus.

   Es werden die Sicherheitsgruppen angezeigt, die in der Region des Ziel-Clusters verfügbar sind, sodass Sie sie auswählen können. Die gewählte Sicherheitsgruppe ist der jeweiligen Verbindung zugeordnet. Weitere Informationen zur Verwendung von Sicherheitsgruppen finden Sie unter Steuern des Datenverkehrs zu Ihren AWS Ressourcen mithilfe von Sicherheitsgruppen im Amazon VPC-Benutzerhandbuch.

   - Rufen Sie sowohl für die regionsübergreifende Replikation (CRR) als auch für die Replikation derselben Region (SRR) die Amazon EC2 EC2-Konsole unter https://console.aws.amazon.com/ec2/ auf und stellen Sie sicher, dass die Sicherheitsgruppen, die Sie dem Replicator zur Verfügung stellen, über ausgehende Regeln verfügen, die den

Datenverkehr zu den Sicherheitsgruppen Ihres Zielclusters zulassen. Stellen Sie außerdem sicher, dass die Sicherheitsgruppen Ihres Ziel-Clusters über Regeln für eingehenden Datenverkehr aus den Replikator-Sicherheitsgruppen verfügen, die für das Ziel bereitgestellt wurden.

So fügen Sie der Sicherheitsgruppe Ihres Zielclusters Regeln für eingehenden Datenverkehr hinzu:

1. Gehen Sie in der AWS Konsole zu den Details Ihres Zielclusters, indem Sie den Clusternamen auswählen.
2. Wählen Sie die Registerkarte Eigenschaften und scrollen Sie dann nach unten zum Bereich Netzwerkeinstellungen, um den Namen der angewendeten Sicherheitsgruppe auszuwählen.
3. Rufen Sie die Regeln für eingehenden Datenverkehr auf und wählen Sie Regeln für eingehenden Datenverkehr bearbeiten.
4. Wählen Sie Regel hinzufügen.
5. Wählen Sie in der Spalte Typ für die neue Regel die Option Benutzerdefiniertes TCP aus.
6. Geben Sie in der Spalte Portbereich den Text ein`9098`. MSK Replicator verwendet IAM-Zugriffskontrolle, um eine Verbindung zu Ihrem Cluster herzustellen, der Port 9098 verwendet.
7. Geben Sie in der Spalte Quelle den Namen der Sicherheitsgruppe ein, die Sie bei der Replikatorerstellung für den Zielcluster angeben werden (dies kann mit der Sicherheitsgruppe des MSK-Zielclusters identisch sein), und wählen Sie dann Regeln speichern aus.

Gehen Sie wie folgt vor, um Regeln für ausgehenden Datenverkehr zur Sicherheitsgruppe von Replicator hinzuzufügen, die für das Ziel bereitgestellt wurde:

1. Gehen Sie in der AWS Konsole zu der Sicherheitsgruppe, die Sie bei der Erstellung des Replikators für das Ziel angeben werden.
2. Wählen Sie die Registerkarte Eigenschaften und scrollen Sie dann nach unten zum Bereich Netzwerkeinstellungen, um den Namen der angewendeten Sicherheitsgruppe auszuwählen.
3. Gehen Sie zu den Regeln für ausgehenden Datenverkehr und wählen Sie Regeln für ausgehenden Datenverkehr bearbeiten aus.
4. Wählen Sie Regel hinzufügen.
5. Wählen Sie in der Spalte Typ für die neue Regel die Option Benutzerdefiniertes TCP aus.

6. Geben Sie in der Spalte Portbereich den Text ein`9098`. MSK Replicator verwendet IAM-Zugriffskontrolle, um eine Verbindung zu Ihrem Cluster herzustellen, der Port 9098 verwendet.
7. Geben Sie in der Spalte Quelle den Namen der Sicherheitsgruppe des MSK-Zielclusters ein, und wählen Sie dann Regeln speichern aus.

> **Note**
>
> Wenn Sie den Datenverkehr nicht mithilfe Ihrer Sicherheitsgruppen einschränken möchten, können Sie alternativ Regeln für eingehenden und ausgehenden Datenverkehr hinzufügen, die den gesamten Datenverkehr zulassen.
> 1. Wählen Sie Regel hinzufügen.
> 2. Wählen Sie in der Spalte Typ die Option Gesamter Datenverkehr aus.
> 3. Geben Sie in der Quelle-Spalte `0.0.0.0/0` ein und wählen Sie dann Regeln speichern.

Einstellungen und Berechtigungen des Replikators konfigurieren

1. Geben Sie im Bereich Replikator-Einstellungen die Themen, die Sie replizieren möchten, mithilfe regulärer Ausdrücke in den Zulassungs- und Verweigerungslisten an. Standardmäßig werden alle Themen repliziert.

   > **Note**
   >
   > MSK Replicator repliziert nur bis zu 750 Themen in sortierter Reihenfolge. Wenn Sie mehr Themen replizieren müssen, empfehlen wir Ihnen, einen separaten Replicator zu erstellen. Rufen Sie das AWS Konsolen-Supportcenter auf und erstellen Sie einen Support-Fall, wenn Sie Support für mehr als 750 Themen pro Replicator benötigen. Sie können die Anzahl der replizierten Themen mithilfe der Metrik "TopicCount" überwachen. Siehe Amazon-MSK-Kontingent.

2. Standardmäßig startet MSK Replicator die Replikation ab dem letzten (neuesten) Offset in den ausgewählten Themen. Alternativ können Sie die Replikation am frühesten (ältesten) Offset in den ausgewählten Themen starten, wenn Sie vorhandene Daten zu Ihren Themen replizieren möchten. Sobald der Replikator erstellt wurde, können Sie diese Einstellung

nicht mehr ändern. Diese Einstellung entspricht dem `startingPosition`Feld in den `CreateReplicator`Anfrage- und `DescribeReplicator`Antwort-APIs.

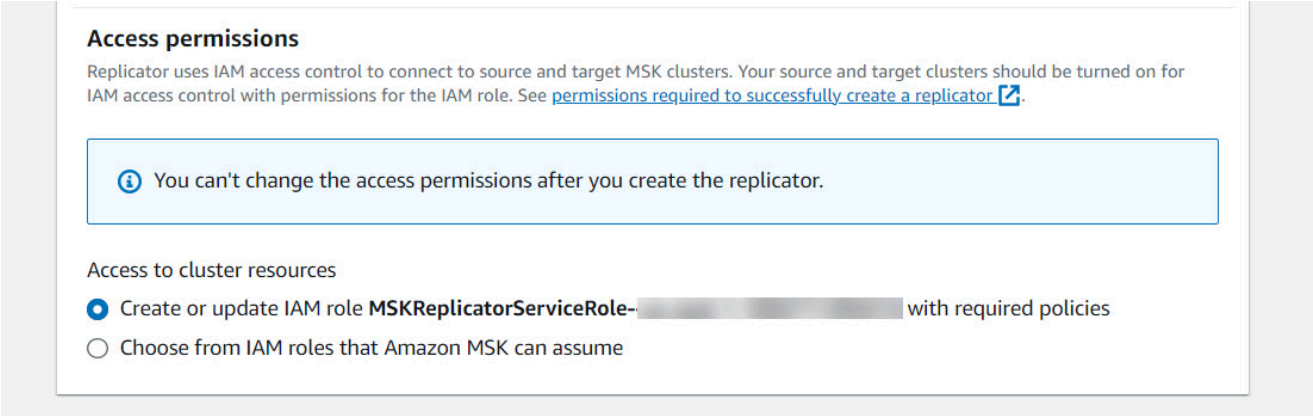
> **Note**
>
> MSK Replicator fungiert wie ein neuer Verbraucher für Ihren Quellcluster. Abhängig von der Datenmenge, die Sie replizieren, und der Kapazität, die Sie in Ihrem Quell-Cluster nutzen, kann dies dazu führen, dass andere Verbraucher in Ihrem Quell-Cluster gedrosselt werden. Wenn Sie einen Replikator erstellen, der auf die früheste Startposition eingestellt ist, liest MSK Replicator zu Beginn eine Datenmenge, die möglicherweise die gesamte verbrauchte Kapazität Ihres Quell-Clusters verbraucht. Sobald Ihr Replicator aufgeholt hat, sollte die Nutzungsrate sinken, um dem Durchsatz Ihrer Quell-Cluster-Themen zu entsprechen. Wenn Sie von Anfang an replizieren, empfehlen wir Ihnen, den Replicator-Durchsatz mithilfe von Kafka-Kontingenten zu verwalten, um sicherzustellen, dass andere Nutzer nicht gedrosselt werden.

3. Standardmäßig kopiert MSK-Replikator alle Metadaten, einschließlich Themenkonfigurationen, Zugriffssteuerungslisten (ACLs) und Verbrauchergruppen-Offsets, um einen reibungslosen Failover zu gewährleisten. Wenn Sie den Replikator nicht für Failover erstellen, können Sie optional eine oder mehrere dieser Einstellungen deaktivieren, die im Abschnitt Zusätzliche Einstellungen verfügbar sind.

> **Note**
>
> MSK-Replikator repliziert keine Schreib-ACLs, da Ihre Produzenten nicht direkt in das replizierte Thema im Ziel-Cluster schreiben sollten. Ihre Produzenten sollten nach dem Failover in das lokale Thema im Ziel-Cluster schreiben. Details dazu finden Sie unter Durchführung eines geplanten Failovers in die sekundäre Region AWS.

4. Geben Sie im Bereich für die Replikation von Verbrauchergruppen die Verbrauchergruppen, die Sie replizieren möchten, mithilfe regulärer Ausdrücke in den Zulassungs- und Verweigerungslisten an. Standardmäßig werden alle Verbrauchergruppen repliziert.
5. Im Bereich Komprimierung können Sie optional wählen, ob die in den Ziel-Cluster geschriebenen Daten komprimiert werden sollen. Wenn Sie die Komprimierung verwenden möchten, empfehlen wir, dieselbe Komprimierungsmethode zu verwenden, wie für die Daten in Ihrem Quell-Cluster.
6. Führen Sie im Bereich Zugriffsberechtigungen einen der folgenden Schritte aus:

a. Wählen Sie IAM-Rolle mit den erforderlichen Richtlinien erstellen oder aktualisieren aus. Die MSK-Konsole hängt der Service-Ausführungsrolle, die für Lese- und Schreibvorgänge in den Quell- und Ziel-MSK-Clustern erforderlich ist, automatisch die erforderlichen Berechtigungen und Vertrauensrichtlinien an.



b. Geben Sie Ihre eigene IAM-Rolle an, indem Sie Aus IAM-Rollen auswählen, die Amazon MSK übernehmen kann auswählen. Wir empfehlen, dass Sie die `AWSMSKReplicatorExecutionRole` verwaltete IAM-Richtlinie Ihrer Rolle für die Serviceausführung zuordnen, anstatt Ihre eigene IAM-Richtlinie zu schreiben.

- Erstellen Sie die IAM-Rolle, die der Replikator zum Lesen und Schreiben in Ihre Quell- und Ziel-MSK-Cluster verwenden wird. Verwenden Sie dabei das unten stehende JSON als Teil der Vertrauensrichtlinie und der der Rolle angehängten `AWSMSKReplicatorExecutionRole`. Ersetzen Sie in der Vertrauensrichtlinie den Platzhalter <yourAccountID> mit Ihrer tatsächlichen Konto-ID.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "kafka.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<yourAccountID>"
                }
            }
```

```
        }
    ]
}
```

7. Im Bereich Replikator-Tags können Sie der MSK-Replikator-Ressource optional Tags zuweisen. Weitere Informationen finden Sie unter Markieren eines Amazon-MSK-Clusters. Bei einem regionsübergreifenden MSK-Replikator werden Tags automatisch mit der Fernregion synchronisiert, wenn der Replikator erstellt wird. Wenn Sie Tags ändern, nachdem der Replikator erstellt wurde, wird die Änderung nicht automatisch mit der Fernregion synchronisiert, sodass Sie lokale Replikator- und Remote-Replikator-Referenzen manuell synchronisieren müssen.
8. Wählen Sie Erstellen aus.

Informationen zum Einschränken von `kafka-cluster:WriteData` Berechtigungen finden Sie im Abschnitt Autorisierungsrichtlinien erstellen unter So funktioniert die IAM-Zugriffskontrolle für Amazon MSK. Sie müssen sowohl dem Quell- als auch dem Zielcluster `kafka-cluster:WriteDataIdempotently` Berechtigungen hinzufügen.

Es dauert ungefähr 30 Minuten, bis der MSK-Replikator erfolgreich erstellt und in den Status RUNNING gewechselt ist.

Wenn Sie einen neuen MSK-Replikator erstellen, um einen gelöschten zu ersetzen, startet der neue Replikator die Replikation ab dem letzten Offset.

Wenn der MSK-Replikator in den Status FAILED übergegangen ist, finden Sie weitere Informationen im Abschnitt Problembehandlung für MSK Replicator.

MSK-Replikator-Einstellungen bearbeiten

Sie können den Quellcluster, den Zielcluster oder die Startposition des Replikators nicht mehr ändern, nachdem der MSK Replicator erstellt wurde. Sie können jedoch andere Replicator-Einstellungen bearbeiten, z. B. Themen und Nutzergruppen, die repliziert werden sollen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon MSK-Konsole unter https://console.aws.amazon.com/msk/home?region=us-east-1#/home/.
2. Wählen Sie im linken Navigationsbereich Replikatoren, um die Liste der Replikatoren im Konto anzuzeigen, und wählen Sie den MSK-Replikator aus, den Sie bearbeiten möchten.
3. Wählen Sie die Registerkarte Eigenschaften aus.
4. Wählen Sie im Bereich Replikator-Einstellungen die Option Replikator bearbeiten.

5. Sie können die MSK-Replikator-Einstellungen bearbeiten, indem Sie eine dieser Einstellungen ändern.

   - Geben Sie die Themen, die Sie replizieren möchten, mithilfe regulärer Ausdrücke in den Zulassungs- und Verweigerungslisten an. Standardmäßig kopiert MSK-Replikator alle Metadaten, einschließlich Themenkonfigurationen, Zugriffssteuerungslisten (ACLs) und Verbrauchergruppen-Offsets, um einen reibungslosen Failover zu gewährleisten. Wenn Sie den Replikator nicht für Failover erstellen, können Sie optional eine oder mehrere dieser Einstellungen deaktivieren, die im Abschnitt Zusätzliche Einstellungen verfügbar sind.

     > **Note**
     >
     > MSK-Replikator repliziert keine Schreib-ACLs, da Ihre Produzenten nicht direkt in das replizierte Thema im Ziel-Cluster schreiben sollten. Ihre Produzenten sollten nach dem Failover in das lokale Thema im Ziel-Cluster schreiben. Details dazu finden Sie unter Durchführung eines geplanten Failovers in die sekundäre Region AWS.

   - Für die Replikation von Verbrauchergruppen können Sie die Verbrauchergruppen, die Sie replizieren möchten, mithilfe regulärer Ausdrücke in den Zulassungs- und Verweigerungslisten angeben. Standardmäßig werden alle Verbrauchergruppen repliziert. Wenn die Zulassungs- und Verweigerungslisten leer sind, ist die Replikation von Verbrauchergruppen deaktiviert.
   - Unter Ziel-Komprimierungstyp können Sie wählen, ob die in den Ziel-Cluster geschriebenen Daten komprimiert werden sollen. Wenn Sie die Komprimierung verwenden möchten, empfehlen wir, dieselbe Komprimierungsmethode zu verwenden, wie für die Daten in Ihrem Quell-Cluster.

6. Speichern Sie Ihre Änderungen.

   Es dauert ungefähr 30 Minuten, bis der MSK-Replikator erfolgreich erstellt und in den Betriebszustand versetzt wurde. Wenn der MSK-Replikator in den Status FAILED übergegangen ist, finden Sie weitere Informationen im Abschnitt über Problembehandlung ???.

Löschen eines MSK-Replikators

Möglicherweise müssen Sie einen MSK-Replikator löschen, wenn er nicht erstellt werden kann (Status FAILED). Die Quell- und Ziel-Cluster, die einem MSK-Replikator zugewiesen sind, können nach der Erstellung des MSK-Replikators nicht mehr geändert werden. Sie können einen vorhandenen MSK-Replikator löschen und einen neuen erstellen. Wenn Sie einen neuen MSK-

Replikator erstellen, um einen gelöschten zu ersetzen, startet der neue Replikator die Replikation ab dem letzten Offset.

1. Melden Sie sich in der AWS Region, in der sich Ihr Quell-Cluster befindet AWS Management Console, bei der an und öffnen Sie die Amazon MSK-Konsole unter https://console.aws.amazon.com/msk/home?region=us-east-1#/home/.
2. Wählen Sie im Navigationsbereich Replikatoren.
3. Wählen Sie aus der Liste der MSK-Replikatoren den Replikator aus, den Sie löschen möchten, und wählen Sie Löschen.

Überwachung einer Replikation

Sie können https://console.aws.amazon.com/cloudwatch/ in der Ziel-Cluster-Region verwenden, um Metriken für `ReplicationLatency`, `MessageLag` und `ReplicatorThroughput` auf Themen- und Aggregatebene für jeden Amazon MSK Replicator anzuzeigen. Metriken sind unter dem ReplicatorNameNamespace „AWS/Kafka“ sichtbar. Sie können auch `ReplicatorFailure`-, `AuthError`- und `ThrottleTime`-Metriken sehen, um nach Problemen zu suchen.

Die MSK-Konsole zeigt eine Teilmenge von CloudWatch Metriken für jeden MSK-Replikator an. Wählen Sie aus der Liste der Replikatoren in der Konsole den Namen eines Replikators aus und wählen Sie die Registerkarte Überwachung.

MSK-Replikatormetriken

Die folgenden Metriken beschreiben Leistungs- oder Verbindungsmetriken für den MSK-Replikator.

AuthError Die Metriken decken keine Authentifizierungsfehler auf Themenebene ab. Um die Authentifizierungsfehler Ihres MSK Replicators auf Themenebene zu überwachen, überwachen Sie die Metriken von Replicator und die ReplicationLatency Metriken des Quellclusters auf Themenebene,. MessagesInPerSec Wenn ein Thema auf 0 ReplicationLatency zurückgesetzt wird, für das Thema aber immer noch Daten erstellt werden, deutet dies darauf hin, dass der Replicator ein Authentifizierungsproblem mit dem Thema hat. Vergewissern Sie sich, dass die IAM-Rolle für die Service-Ausführungsrolle des Replikators über ausreichende Berechtigungen für den Zugriff auf das Thema verfügt.

Metriktyp	Metrik	Beschreibung	Dimensionen	Einheit	Granularität unformatierter Metriken	Aggregationsstatistik unformatierter Metriken
Leistung	ReplicationLatency	Zeit, die für die Replikation von Datensätzen vom Quell- zum Ziel-Cluster benötigt wird; Dauer zwischen der Produktionszeit von Datensätzen an der Quelle und der Replikation zum Ziel. Wenn ReplicationLatency die Zahl steigt, überprüfen Sie, ob die Cluster über genügend Partitionen verfügen, um die Replikation zu unterstützen. Eine hohe Replikationslatenz kann auftreten, wenn die Anzahl der Partitionen für	ReplicatorName	Millisekunden	Partition	Maximum
			ReplicatorName, Thema	Millisekunden	Partition	Maximum

Metriktyp	Metrik	Beschreibung	Dimensionen	Einheit	Granularität unformatierter Metriken	Aggregationsstatistik unformatierter Metriken	
		einen hohen Durchsatz zu niedrig ist.					

Metriktyp	Metrik	Beschreibung	Dimensionen	Einheit	Granularität unformatierter Metriken	Aggregationsstatistik unformatierter Metriken
Leistung	MessageLag	Überwacht die Synchronisation zwischen dem MSK Replicator und dem Quellcluster. MessageLag gibt die Verzögerung zwischen den Nachrichten, die an den Quellcluster gesendet werden, und den Nachrichten, die vom Replikator verarbeitet werden, an. Es ist nicht die Verzögerung zwischen dem Quell- und dem Zielcluster. Selbst wenn der Quellcluster nicht verfügbar oder unterbrochen ist, beendet der Replikator das Schreiben	ReplicatorName	Anzahl	Partition	Summe
			ReplicatorName, Thema	Anzahl	Partition	Summe

Metriktyp	Metrik	Beschreibung	Dimensionen	Einheit	Granularität unformatierter Metriken	Aggregationsstatistik unformatierter Metriken	
		der verbrauchten Nachricht in den Zielcluster. MessageLag Zeigt nach einem Ausfall einen Anstieg an, der die Anzahl der Nachrichten angibt, die der Replikator hinter dem Quellcluster hat. Diese Zahl kann überwacht werden, bis die Anzahl der Nachrichten 0 ist, was bedeutet, dass der Replikator den Quellcluster eingeholt hat.					

Metriktyp	Metrik	Beschreibung	Dimensionen	Einheit	Granularität unformatierter Metriken	Aggregationsstatistik unformatierter Metriken	
Leistung	ReplicatorThroughput	Durchschnittliche Anzahl der pro Sekunde replizierten Bytes. Falls ReplicatorThroughput ein Thema angezeigt wird, überprüfen Sie KafkaClusterPingSuccessCount anhand von AuthError Metriken, ob der Replicator mit Clustern kommunizieren kann. Überprüfen Sie anschließend die Cluster-Metriken, um sicherzustellen, dass der Cluster nicht ausgefallen ist.	ReplicatorName	BytesPerSecond	Partition	Summe	
			ReplicatorName, Thema	BytesPerSecond	Partition	Summe	

Metriktyp	Metrik	Beschreibung	Dimensionen	Einheit	Granularität unformatierter Metriken	Aggregationsstatistik unformatierter Metriken
Debugge	AuthError	Die Anzahl der Verbindungen mit fehlgeschlagener Authentifizierung pro Sekunde. Wenn diese Metrik über 0 liegt, können Sie überprüfen, ob die Richtlinie der Service-Ausführungsrolle für den Replikator gültig ist, und sicherstellen, dass für die Cluster-Berechtigungen keine Verweigerungs-Berechtigungen festgelegt sind. Anhand der clusterAlias-Dimension können Sie feststellen, ob im Quell- oder Ziel-Cluster Authentif	ReplicatorName, ClusterAlias	Anzahl	Worker	Summe

Metriktyp	Metrik	Beschreibung	Dimensionen	Einheit	Granularität unformatierter Metriken	Aggregationsstatistik unformatierter Metriken	
		izierungsfehler auftreten.					

Metriktyp	Metrik	Beschreibung	Dimensionen	Einheit	Granularität unformatierter Metriken	Aggregationsstatistik unformatierter Metriken
Debugge	ThrottleTime	Die durchschnittliche Zeit in ms, in der eine Anfrage von Brokern im Cluster gedrosselt wurde. Stellen Sie die Drosselung ein, um zu verhindern, dass der MSK-Replikator den Cluster überlastet. Wenn diese Metrik 0 ist, ReplicationLatency nicht hoch ist und ReplicatorThroughput erwartungsgemäß ist, dann funktioniert die Drosselung erwartungsgemäß. Wenn diese Metrik über 0 liegt,	ReplicatorName, ClusterAlias	Millisekunden	Worker	Maximum

Metriktyp	Metrik	Beschreibung	Dimensio en	Einheit	Granular tät unformat erter Metriken	Aggrega onsstatis tik unformat erter Metriken
		können Sie die Drosselung entsprechend anpassen.				
Debugge	ReplicatorFailure	Anzahl der Fehler, die beim Replikator auftreten.	Replicato rName	Anzahl		Summe

Metriktyp	Metrik	Beschreibung	Dimensionen	Einheit	Granularität unformatierter Metriken	Aggregationsstatistik unformatierter Metriken
Debugge	KafkaClusterPingSuccessCount	Zeigt den Zustand der Replikatorverbindung zum Kafka-Cluster an. Wenn dieser Wert 1 ist, ist die Verbindung fehlerfrei. Wenn der Wert 0 oder kein Datenpunkt ist, ist die Verbindung fehlerhaft. Wenn der Wert 0 ist, können Sie die Netzwerk- oder IAM-Berechtigungseinstellungen für den Kafka-Cluster überprüfen. Anhand der ClusterAlias Dimension können Sie feststellen, ob diese Metrik für den Quell-	ReplicatorName, ClusterAlias	Anzahl		Summe

Metriktyp	Metrik	Beschreibung	Dimensionen	Einheit	Granularität unformatierter Metriken	Aggregationsstatistik unformatierter Metriken	
		oder Zielcluster bestimmt ist.					

Verwendung von Replikation zur Erhöhung der Ausfallsicherheit einer Kafka-Streaming-Anwendung in allen Regionen

Sie können MSK Replicator verwenden, um Aktiv-Aktiv- oder Aktiv-Passiv-Cluster-Topologien einzurichten, um die Resilienz Ihrer Apache Kafka-Anwendung in allen Regionen zu erhöhen. AWS In einer aktiv-aktiven Einrichtung verarbeiten beide MSK-Cluster aktiv Lese- und Schreibvorgänge. In einer Aktiv-Passiv-Einrichtung stellt jeweils nur ein MSK-Cluster aktiv Streaming-Daten bereit, während sich der andere Cluster im Standby-Modus befindet.

Überlegungen zur Erstellung von Apache-Kafka-Anwendungen mit mehreren Regionen

Ihre Verbraucher müssen in der Lage sein, doppelte Nachrichten ohne nachgelagerte Auswirkungen erneut zu verarbeiten. MSK Replicator repliziert Daten, was zu Duplikaten im Standby-Cluster führen kann. at-least-once Wenn Sie zur sekundären AWS Region wechseln, verarbeiten Ihre Kunden dieselben Daten möglicherweise mehrmals. MSK-Replikator räumt dem Kopieren von Daten Vorrang vor Verbraucher-Offsets ein, um eine bessere Leistung zu erzielen. Nach einem Failover kann der Verbraucher beginnen, aus früheren Offsets zu lesen, was zu einer doppelten Verarbeitung führt.

Produzenten und Verbraucher müssen auch den Verlust minimaler Daten hinnehmen. Da MSK Replicator Daten asynchron repliziert, kann nicht garantiert werden, dass alle Daten AWS in die sekundäre Region repliziert werden, wenn es in der primären Region zu Ausfällen kommt. Sie können die Replikationslatenz verwenden, um die maximale Anzahl von Daten zu ermitteln, die nicht in die sekundäre Region kopiert wurden.

Verwendung einer Aktiv-Aktiv-Cluster-Topologie im Vergleich zur Aktiv-Passiv-Cluster-Topologie

Eine Aktiv-Aktiv-Cluster-Topologie bietet eine Wiederherstellungszeit von nahezu Null und ermöglicht den gleichzeitigen Betrieb Ihrer Streaming-Anwendung in mehreren AWS -Regionen. Wenn ein

Cluster in einer Region beeinträchtigt ist, verarbeiten Anwendungen, die mit dem Cluster in der anderen Region verbunden sind, weiterhin Daten.

Aktiv-Passiv-Einrichtungen eignen sich für Anwendungen, die jeweils nur in einer AWS - Region ausgeführt werden können, oder wenn Sie mehr Kontrolle über die Reihenfolge der Datenverarbeitung benötigen. Aktiv-Passiv-Einrichtungen benötigen mehr Wiederherstellungszeit als Aktiv-Aktiv-Einrichtungen, da Sie Ihre gesamte Aktiv-Passiv-Einrichtung, einschließlich der Produzenten und Verbraucher, in der sekundären Region starten müssen, um das Streamen von Daten nach einem Failover wieder aufnehmen zu können.

Erstellen einer Aktiv-Passiv-Cluster-Einrichtung für Kafka und Benennung replizierter Themen

Für ein Aktiv-Passiv-Setup empfehlen wir Ihnen, eine ähnliche Konfiguration von Produzenten, MSK-Clustern und Verbrauchern (mit demselben Verbrauchergruppennamen) in zwei verschiedenen Regionen zu betreiben. AWS Es ist wichtig, dass die beiden MSK-Cluster über identische Lese- und Schreibkapazitäten verfügen, um eine zuverlässige Datenreplikation zu gewährleisten. Sie müssen einen MSK-Replikator erstellen, um kontinuierlich Daten vom primären Cluster auf den Standby-Cluster zu kopieren. Sie müssen Ihre Producer auch so konfigurieren, dass sie Daten in Themen eines Clusters in derselben Region schreiben. AWS

Um sicherzustellen, dass Ihre Verbraucher die Verarbeitung zuverlässig vom Standby-Cluster aus wieder aufnehmen können, müssen Sie Ihre Verbraucher so konfigurieren, dass sie Daten aus den Themen mithilfe des Platzhalteroperators „.*“ lesen. MSK Replicator repliziert beispielsweise „topic1“ aus dem primären Cluster auf ein neues Thema im Standby-Cluster namens „**<** alias**>**.topic1“. sourceKafkaCluster Beispielsweise können Sie in beiden Regionen Ihre Produzenten so konfigurieren, dass sie in „topic1“ schreiben und Ihre Verbraucher, dass sie von „.*topic1“ verbrauchen. Dieses Beispiel würde auch ein Thema wie footopic1 beinhalten. Passen Sie den Platzhalteroperator also an Ihre Bedürfnisse an.

Wann sollte ein AWS Failover zur sekundären Region durchgeführt werden

Wir empfehlen, dass Sie die Replikationslatenz in der sekundären AWS Region mithilfe von CloudWatch überwachen. Während eines Serviceereignisses in der primären AWS Region kann die Replikationslatenz plötzlich ansteigen. Wenn die Latenz weiter zunimmt, verwenden Sie das AWS Service Health Dashboard, um nach Serviceereignissen in der primären AWS Region zu suchen. Wenn ein Ereignis eintritt, können Sie ein Failover auf die sekundäre AWS Region durchführen.

Durchführung eines geplanten Failovers in die sekundäre Region AWS

Sie können einen geplanten Failover durchführen, um die Widerstandsfähigkeit Ihrer Anwendung gegen ein unerwartetes Ereignis in Ihrer primären AWS Region zu testen, in der sich Ihr MSK-Quellcluster befindet. Ein geplantes Failover sollte nicht zu Datenverlust führen.

1. Fahren Sie alle Produzenten und Verbraucher herunter, die eine Verbindung zum Quell-Cluster herstellen.
2. Erstellen Sie einen neuen MSK-Replikator, um Daten aus Ihrem MSK-Cluster in der sekundären Region auf Ihren MSK-Cluster in der primären Region zu replizieren. Dies ist erforderlich, um die Daten, die Sie in die sekundäre Region schreiben werden, zurück in die primäre Region zu kopieren, sodass Sie nach dem Ende des unerwarteten Ereignisses ein Failback zur primären Region durchführen können.
3. Starten Sie die Produzenten auf dem Zielcluster in der sekundären AWS Region.
4. Befolgen Sie die Schritte auf einer der folgenden Registerkarten, je nachdem, welche Anforderungen Ihre Anwendung für die Nachrichtenreihenfolge hat.

No message ordering

Wenn für Ihre Anwendung keine Nachrichtenreihenfolge erforderlich ist, starten Sie Benutzer in der sekundären AWS Region, die sowohl aus dem lokalen (z. B.`topic`) als auch aus dem replizierten Thema (z. B.`<sourceKafkaClusterAlias>.topic`) lesen, mithilfe eines Platzhalteroperators (z. B.). *Thema).

Message ordering

Wenn Ihre Anwendung eine Nachrichtenreihenfolge erfordert, starten Sie Verbraucher nur für die replizierten Themen auf dem Ziel-Cluster (z. B. `<sourceKafkaClusterAlias>.topic`), aber nicht für die lokalen Themen (z. B. `topic`).

1. Warten Sie, bis alle Verbraucher replizierter Themen auf dem Ziel-MSK-Cluster die Verarbeitung aller Daten abgeschlossen haben, sodass die Verbraucherverzögerung 0 und die Anzahl der verarbeiteten Datensätze ebenfalls 0 ist. Stoppen Sie dann die Verbraucher für die replizierten Themen auf dem Ziel-Cluster. Zu diesem Zeitpunkt sind alle Datensätze, die vom Quell-MSK-Cluster auf den Ziel-MSK-Cluster repliziert wurden, verbraucht.
2. Starten Sie die Verbraucher für die lokalen Themen (z. B. `topic`) auf dem Ziel-MSK-Cluster.

Durchführung eines ungeplanten Failovers in die sekundäre Region AWS

Sie können einen ungeplanten Failover durchführen, wenn in der primären AWS Region, in der sich Ihr Quell-MSK-Cluster befindet, ein Serviceereignis auftritt und Sie Ihren Datenverkehr vorübergehend in die sekundäre AWS Region umleiten möchten, in der sich Ihr Ziel-MSK-Cluster befindet. Ein ungeplanter Failover kann zu Datenverlusten führen.

1. Versuchen Sie, alle Produzenten und Verbraucher, die in der primären Region eine Verbindung zum MSK-Quell-Cluster herstellen, herunterzufahren. Dies könnte fehlschlagen.
2. Starten Sie Produzenten, die eine Verbindung zum Ziel-Cluster in der sekundären Region herstellen.
3. Befolgen Sie die Schritte auf einer der folgenden Registerkarten, je nachdem, welche Anforderungen Ihre Anwendung für die Nachrichtenreihenfolge hat.

   No message ordering

   Wenn für Ihre Anwendung keine Nachrichtenreihenfolge erforderlich ist, starten Sie Benutzer in der AWS Zielregion, die sowohl aus lokalen (z. B.) als auch aus replizierten Themen (z. B.`topic`) lesen, indem Sie einen Platzhalteroperator (z. B.`<sourceKafkaClusterAlias>.topic`) verwenden. `.*topic`

   Message ordering

   1. Starten Sie Verbraucher nur für die replizierten Themen auf dem Ziel-Cluster (z. B. `<sourceKafkaClusterAlias>.topic`), aber nicht für die lokalen Themen (z. B. `topic`).
   2. Warten Sie, bis alle Verbraucher replizierter Themen auf dem Ziel-MSK-Cluster die Verarbeitung aller Daten abgeschlossen haben, sodass die Offset-Verzögerung 0 und die Anzahl der verarbeiteten Datensätze ebenfalls 0 ist. Stoppen Sie dann die Verbraucher für die replizierten Themen auf dem Ziel-Cluster. Zu diesem Zeitpunkt sind alle Datensätze, die vom Quell-MSK-Cluster auf den Ziel-MSK-Cluster repliziert wurden, verbraucht.
   3. Starten Sie die Verbraucher für die lokalen Themen (z. B. `topic`) auf dem Ziel-MSK-Cluster.
4. Sobald das Serviceereignis in der primären Region beendet ist, erstellen Sie einen neuen MSK-Replikator, um Daten von Ihrem MSK-Cluster in der sekundären Region auf Ihren MSK-Cluster in der primären Region zu replizieren, wobei die Replicator-Startposition auf „Early“ gesetzt ist. Dies ist erforderlich, um die Daten, die Sie in die sekundäre Region schreiben werden, zurück in die primäre Region zu kopieren, sodass Sie nach dem Ende des Service-Ereignisses ein Failback

zur primären Region durchführen können. Wenn Sie die Replicator-Startposition nicht auf „Early“ setzen, werden alle Daten, die Sie während des Serviceereignisses in der primären Region für den Cluster in der sekundären Region erzeugt haben, nicht zurück in den Cluster in der primären Region kopiert.

Ein Failback zur primären Region wird durchgeführt AWS

Sie können ein Failback zur primären AWS Region durchführen, nachdem das Serviceereignis in dieser Region beendet ist. MSK-Replikator überspringt automatisch Themen, die den Quell-Cluster-Alias als Präfix haben, wenn Daten während eines Failbacks zurück in die primäre Region repliziert werden.

Wenn Sie die Schritte für das ungeplante Failover befolgt haben, sollten Sie den Failback-Replikator bereits als Teil des letzten Schritts des Failovers von der primären zur sekundären Region erstellt haben.

Wenn Sie die ungeplanten Failover-Schritte nicht befolgt haben, erstellen Sie nach dem Ende des Serviceereignisses in der primären Region einen neuen MSK-Replikator, um Daten von Ihrem MSK-Cluster in der sekundären Region auf Ihren MSK-Cluster in der primären Region zu replizieren, wobei die Replicator-Startposition auf „Early“ gesetzt ist. Dies ist erforderlich, um die Daten, die Sie in die sekundäre Region schreiben werden, zurück in die primäre Region zu kopieren, sodass Sie nach dem Ende des Service-Ereignisses ein Failback zur primären Region durchführen können. Wenn Sie die Startposition des Replikators nicht vom Standardwert „Späteste“ auf „Früheste“ ändern, werden alle Daten, die Sie während des Serviceereignisses in der primären Region für den Cluster in der sekundären Region erzeugt haben, nicht zurück in den Cluster in der primären Region kopiert.

Sie sollten Failback-Schritte erst einleiten, wenn die Replikation vom Cluster in der sekundären Region zum Cluster in der primären Region abgeschlossen ist und die MessageLag Metrik in fast CloudWatch 0 liegt. Ein geplantes Failback sollte nicht zu Datenverlust führen.

1. Fahren Sie alle Produzenten und Verbraucher herunter, die in der sekundären Region eine Verbindung zum MSK-Cluster herstellen.
2. Löschen Sie bei einer Aktiv-Passiv-Topologie den Replikator, der Daten aus dem Cluster in der sekundären Region in die primäre Region repliziert. Sie müssen den Replikator für eine Aktiv-Aktiv-Topologie nicht löschen.
3. Starten Sie Produzenten, die eine Verbindung zum MSK-Cluster in der primären Region herstellen.

4. Befolgen Sie die Schritte auf einer der folgenden Registerkarten, je nachdem, welche Anforderungen Ihre Anwendung für die Nachrichtenreihenfolge hat.

   No message ordering

   Wenn für Ihre Anwendung keine Nachrichtenreihenfolge erforderlich ist, starten Sie Benutzer in der primären AWS Region, die sowohl aus dem lokalen (z. B.`topic`) als auch aus den replizierten Themen (z. B.`<sourceKafkaClusterAlias>.topic`) lesen, indem Sie einen Platzhalteroperator (z. B.) verwenden. `.*topic` Die Verbraucher, die sich mit lokalen Themen (z. B. Thema) befassen, beginnen ab dem letzten Offset, das sie vor dem Failover konsumiert haben. Wenn vor dem Failover unverarbeitete Daten vorhanden waren, werden diese jetzt verarbeitet. Im Falle eines geplanten Failovers sollte es keinen solchen Datensatz geben.

   Message ordering

   1. Starten Sie Verbraucher nur für die replizierten Themen in der primären Region (z. B. `<sourceKafkaClusterAlias>.topic`), aber nicht für die lokalen Themen (z. B. `topic`).
   2. Warten Sie, bis alle Verbraucher replizierter Themen auf dem Cluster in der primären Region die Verarbeitung aller Daten abgeschlossen haben, sodass die Offset-Verzögerung 0 und die Anzahl der verarbeiteten Datensätze ebenfalls 0 ist. Stoppen Sie dann die Verbraucher für die replizierten Themen auf dem Cluster in der primären Region. Zu diesem Zeitpunkt wurden alle Datensätze, die nach dem Failover in der sekundären Region erstellt wurden, in der primären Region verbraucht.
   3. Starten Sie Verbraucher für die lokalen Themen (z. B. `topic`) auf dem Cluster in der primären Region.

5. Stellen Sie anhand der Metriken und Latenz sicher, dass sich der bestehende Replikator vom Cluster in der primären Region zum Cluster in der sekundären Region im Status RUNNING befindet und erwartungsgemäß funktioniert. `ReplicatorThroughput`

Erstellen einer Aktiv-Aktiv-Einrichtung mit MSK-Replikator

Gehen Sie wie folgt vor, um eine Aktiv-Aktiv-Topologie zwischen dem Quell-MSK-Cluster A und dem Ziel-MSK-Cluster B einzurichten.

1. Erstellen Sie einen MSK-Replikator mit MSK-Cluster A als Quelle und MSK-Cluster B als Ziel.

2. Nachdem der obige MSK-Replikator erfolgreich erstellt wurde, erstellen Sie einen Replikator mit Cluster B als Quelle und Cluster A als Ziel.
3. Erstellen Sie zwei Gruppen von Produzenten, von denen jeder gleichzeitig Daten in das lokale Thema (z. B. „topic“) im Cluster in derselben Region wie der Produzent schreibt.
4. Erstellen Sie zwei Gruppen von Verbrauchern, die jeweils Daten mithilfe eines Wildcard-Abonnements lesen (z. B.“). *topic“) aus dem MSK-Cluster in derselben AWS Region wie der Verbraucher. Auf diese Weise lesen Ihre Verbraucher automatisch Daten, die lokal in der Region erzeugt wurden, aus dem lokalen Thema (z. B. `topic`) sowie Daten, die aus einer anderen Region repliziert wurden im Thema mit dem Präfix `<sourceKafkaClusterAlias>.topic`. Diese beiden Gruppen von Verbrauchern sollten unterschiedliche Verbrauchergruppen-IDs haben, damit die Verbrauchergruppen-Offsets nicht überschrieben werden, wenn MSK Replicator sie in den anderen Cluster kopiert.

Fehlerbehebung für MSK-Replikator

Themen

Die folgenden Informationen können zum Beheben von Problemen mit MSK-Replikator nützlich sein. Sie können Ihr Problem auch im AWS re:Post posten.

Der Status des MSK-Replikators wechselt von CREATING zu FAILED

Im Folgenden sind einige der häufigsten Ursachen für Fehler bei der Erstellung des MSK-Replikators aufgeführt.

1. Stellen Sie sicher, dass die Sicherheitsgruppen, die Sie für den Replikator im Ziel-Cluster-Bereich angeben, über Regeln für ausgehenden Datenverkehr zu den Sicherheitsgruppen Ihres Ziel-Clusters verfügen. Stellen Sie außerdem sicher, dass die Sicherheitsgruppen Ihres Ziel-Clusters

über Regeln für eingehenden Datenverkehr aus den Sicherheitsgruppen verfügen, die Sie im Ziel-Cluster-Bereich für die Replikator-Erstellung bereitstellen. Siehe Wählen Sie den Ziel-Cluster.

2. Wenn Sie einen Replikator für die regionsübergreifende Replikation erstellen, stellen Sie sicher, dass in Ihrem Quell-Cluster Multi-VPC-Konnektivität für die IAM-Access-Control-Authentifizierungsmethode aktiviert ist. Siehe Private Multi-VPC-Konnektivität von Amazon MSK in einer einzelnen Region. Stellen Sie außerdem sicher, dass die Cluster-Richtlinie auf dem Quell-Cluster eingerichtet ist, sodass der MSK-Replikator eine Verbindung zum Quell-Cluster herstellen kann. Siehe Schritt 1: Den Amazon-MSK-Quell-Cluster vorbereiten.
3. Stellen Sie sicher, dass die IAM-Rolle, die Sie bei der Erstellung des MSK-Replikators angegeben haben, über die erforderlichen Berechtigungen zum Lesen und Schreiben in die Quell- und Ziel-Cluster verfügt. Stellen Sie außerdem sicher, dass die IAM-Rolle über Schreibberechtigungen für Themen verfügt. Siehe Einstellungen und Berechtigungen des Replikators konfigurieren
4. Stellen Sie sicher, dass Ihre Netzwerk-ACLs die Verbindung zwischen dem MSK-Replikator und Ihren Quell- und Ziel-Clustern nicht blockieren.
5. Es ist möglich, dass Quell- oder Ziel-Cluster nicht vollständig verfügbar waren, als der MSK-Replikator versucht hat, eine Verbindung zu ihnen herzustellen. Dies kann auf eine übermäßige Last, Festplattennutzung oder CPU-Auslastung zurückzuführen sein, wodurch der Replikator keine Verbindung zu den Brokern herstellen kann. Beheben Sie das Problem mit den Brokern und versuchen Sie erneut, den Replikator zu erstellen.

Nachdem Sie die oben genannten Validierungen durchgeführt haben, erstellen Sie den MSK-Replikator erneut.

Der MSK-Replikator scheint im Status CREATING festzustecken

Gelegentlich dauert die MSK-Replikator-Erstellung bis zu 30 Minuten. Warten Sie 30 Minuten und überprüfen Sie den Status des Replikators erneut.

Der MSK-Replikator repliziert keine Daten oder repliziert nur Teildaten

Gehen Sie wie folgt vor, um Probleme bei der Datenreplikation zu beheben.

1. Vergewissern Sie sich anhand der von MSK Replicator unter bereitgestellten AuthError Metrik, dass Ihr Replicator keine Authentifizierungsfehler aufweist. CloudWatch Wenn diese Metrik über 0 liegt, können Sie überprüfen, ob die Richtlinie der IAM-Rolle für den Replikator gültig ist, und sicherstellen, dass für die Cluster-Berechtigungen keine Verweigerungs-Berechtigungen festgelegt

sind. Anhand der clusterAlias-Dimension können Sie feststellen, ob im Quell- oder Ziel-Cluster Authentifizierungsfehler auftreten.

2. Stellen Sie sicher, dass bei Ihren Quell- und Ziel-Clustern keine Probleme auftreten. Es ist möglich, dass der Replikator keine Verbindung zu Ihrem Quell- oder Ziel-Cluster herstellen kann. Dies kann auf zu viele Verbindungen, eine voll ausgelastete Festplatte oder eine hohe CPU-Auslastung zurückzuführen sein.
3. Stellen Sie mithilfe der Metrik in sicher, dass Ihre Quell- und Zielcluster von MSK Replicator aus erreichbar sind. KafkaClusterPingSuccessCount CloudWatch Anhand der clusterAlias-Dimension können Sie feststellen, ob im Quell- oder Ziel-Cluster Authentifizierungsfehler auftreten. Wenn diese Metrik 0 ist oder keinen Datenpunkt hat, ist die Verbindung fehlerhaft. Sie sollten die Netzwerk- und IAM-Rollenberechtigungen überprüfen, die MSK-Replikator für die Verbindung mit Ihren Clustern verwendet.
4. Stellen Sie anhand der Metrik in sicher, dass Ihr Replicator nicht aufgrund fehlender Berechtigungen auf Themenebene ausfällt. ReplicatorFailure CloudWatch Wenn diese Metrik über 0 liegt, überprüfen Sie die von Ihnen angegebene IAM-Rolle für Berechtigungen auf Themenebene.
5. Vergewissern Sie sich, dass der reguläre Ausdruck, den Sie bei der Erstellung des Replikators in der Zulassungsliste angegeben haben, mit den Namen der Themen übereinstimmt, die Sie replizieren möchten. Stellen Sie außerdem sicher, dass die Themen nicht aufgrund eines regulären Ausdrucks in der Verweigerungsliste von der Replikation ausgeschlossen werden.
6. Beachten Sie, dass es bis zu 30 Sekunden dauern kann, bis der Replicator die neuen Themen oder Themenpartitionen auf dem Zielcluster erkennt und erstellt. Alle Nachrichten, die an das Quellthema gesendet wurden, bevor das Thema auf dem Zielcluster erstellt wurde, werden nicht repliziert, wenn der Replikator die neueste Startposition hat (Standard). Alternativ können Sie die Replikation auch an der frühesten Stelle in den Themenpartitionen des Quellclusters starten, wenn Sie vorhandene Nachrichten zu Ihren Themen auf dem Zielcluster replizieren möchten. Siehe Einstellungen und Berechtigungen des Replikators konfigurieren.

Die Nachrichtenoffsets im Zielcluster unterscheiden sich von denen im Quellcluster

Im Rahmen der Datenreplikation verarbeitet MSK Replicator Nachrichten aus dem Quellcluster und sendet sie an den Zielcluster. Dies kann dazu führen, dass Nachrichten auf Ihrem Quell- und Zielcluster unterschiedliche Offsets aufweisen. Wenn Sie jedoch bei der Erstellung des Replikators die Synchronisierung von Offsets für Nutzergruppen aktiviert haben, übersetzt MSK Replicator die

Offsets beim Kopieren der Metadaten automatisch, sodass Ihre Benutzer nach einem Failover zum Zielcluster die Verarbeitung fast dort fortsetzen können, wo sie im Quellcluster aufgehört haben.

MSK Replicator synchronisiert keine Nutzungsgruppen, Offsets oder die Nutzungsgruppe ist auf dem Zielcluster nicht vorhanden

Gehen Sie wie folgt vor, um Probleme mit der Metadatenreplikation zu beheben.

1. Stellen Sie sicher, dass Ihre Datenreplikation wie erwartet funktioniert. Falls nicht, siehe Der MSK-Replikator repliziert keine Daten oder repliziert nur Teildaten.
2. Stellen Sie sicher, dass der reguläre Ausdruck, den Sie bei der Erstellung des Replikators in der Zulassungsliste angegeben haben, mit den Namen der Nutzergruppen übereinstimmt, die Sie replizieren möchten. Stellen Sie außerdem sicher, dass die Nutzergruppen nicht aufgrund eines regulären Ausdrucks in der Ablehnungsliste von der Replikation ausgeschlossen werden.
3. Stellen Sie sicher, dass MSK Replicator das Thema auf dem Zielcluster erstellt hat. Es kann bis zu 30 Sekunden dauern, bis der Replicator die neuen Themen oder Themenpartitionen auf dem Zielcluster erkannt und erstellt hat. Alle Nachrichten, die an das Quellthema gesendet wurden, bevor das Thema auf dem Zielcluster erstellt wurde, werden nicht repliziert, wenn der Replikator die neueste Startposition hat (Standard). Wenn Ihre Nutzergruppe auf dem Quellcluster nur die Nachrichten verwendet hat, die nicht von MSK Replicator repliziert wurden, wird die Nutzungsgruppe nicht auf den Zielcluster repliziert. Nachdem das Thema erfolgreich auf dem Zielcluster erstellt wurde, beginnt MSK Replicator mit der Replikation neu geschriebener Nachrichten auf dem Quellcluster auf das Ziel. Sobald Ihre Nutzergruppe beginnt, diese Nachrichten von der Quelle zu lesen, repliziert MSK Replicator die Nutzergruppe automatisch auf den Zielcluster. Alternativ können Sie die Replikation ab dem frühesten Offset in den Themenpartitionen des Quell-Clusters starten, wenn Sie vorhandene Nachrichten zu Ihren Themen auf dem Zielcluster replizieren möchten. Siehe Einstellungen und Berechtigungen des Replikators konfigurieren.

> **Note**
>
> MSK Replicator optimiert die Offset-Synchronisierung von Nutzergruppen für Ihre Benutzer auf dem Quellcluster, die von einer Position aus lesen, die näher am Ende der Themenpartition liegt. Wenn Ihre Nutzergruppen im Quell-Cluster hinterherhinken, können Sie bei diesen Nutzergruppen auf dem Ziel-Cluster eine höhere Verzögerung feststellen als beim Quell-Cluster. Das bedeutet, dass Ihre Kunden nach einem Failover auf den Zielcluster

mehr doppelte Nachrichten erneut verarbeiten werden. Um diese Verzögerung zu verringern, müssten Ihre Verbraucher auf dem Quell-Cluster aufholen und von der Spitze des Streams (Ende der Themenpartition) aus mit dem Konsum beginnen. Wenn Ihre Kunden aufholen, reduziert MSK Replicator die Verzögerung automatisch.

Die Replikationslatenz ist hoch oder nimmt weiter zu

Im Folgenden sind einige der häufigsten Ursachen für eine hohe Replikationslatenz aufgeführt.

1. Stellen Sie sicher, dass Sie die richtige Anzahl von Partitionen auf Ihren Quell- und Ziel-MSK-Clustern haben. Zu wenige oder zu viele Partitionen können sich auf die Leistung auswirken. Hinweise zur Auswahl der Anzahl von Partitionen finden Sie unter Bewährte Methoden für die Verwendung von MSK-Replikator. Die folgende Tabelle zeigt die empfohlene Mindestanzahl von Partitionen, um mit MSK-Replikator den gewünschten Durchsatz zu erzielen.

   Durchsatz und empfohlene Mindestanzahl von Partitionen

Durchsatz (MB/s)	Mindestanzahl an Partitionen erforderlich
50	167
100	334
250	833
500	166
1000	3333

2. Stellen Sie sicher, dass Ihre Quell- und Ziel-MSK-Cluster über genügend Lese- und Schreibkapazität verfügen, um den Replikations-Datenverkehr zu unterstützen. MSK-Replikator fungiert als Verbraucher für Ihren Quell-Cluster (Ausgang) und als Produzent für Ihren Ziel-Cluster (Eingang). Daher sollten Sie Cluster-Kapazität bereitstellen, um den Replikations-Datenverkehr zusätzlich zu anderem Datenverkehr auf Ihren Clustern zu unterstützen. Hinweise zur Dimensionierung Ihrer MSK-Cluster finden Sie unter ???.
3. Die Replikationslatenz kann für MSK-Cluster in verschiedenen Quell- und AWS Zielregionspaaren variieren, je nachdem, wie weit die Cluster geografisch voneinander entfernt sind. Beispielsweise ist die Replikationslatenz bei der Replikation zwischen Clustern in den Regionen Europa (Irland)

und Europa (London) in der Regel niedriger als bei der Replikation zwischen Clustern in den Regionen Europa (Irland) und Asien-Pazifik (Sydney).

4. Stellen Sie sicher, dass Ihr Replikator nicht aufgrund zu aggressiver Kontingente auf Ihren Quell- oder Ziel-Clustern gedrosselt wird. Sie können die von MSK Replicator bereitgestellte ThrottleTime Metrik verwenden, um die durchschnittliche Zeit in Millisekunden CloudWatch zu ermitteln, für die eine Anfrage von Brokern auf Ihrem Quell-/Zielcluster gedrosselt wurde. Wenn diese Metrik über 0 liegt, sollten Sie die Kafka-Kontingente anpassen, um die Drosselung zu reduzieren, damit der Replikator aufholen kann. Informationen zur Verwaltung von Kafka-Kontingenten für den Replikator finden Sie unter Verwaltung des MSK-Replikator-Durchsatzes mithilfe von Kafka-Kontingenten.
5. ReplicationLatency und kann zunehmen, wenn eine Region heruntergestuft wird. MessageLag AWS Verwenden Sie das AWS Service Health Dashboard, um in der Region, in der sich Ihr primärer MSK-Cluster befindet, nach einem MSK-Service-Ereignis zu suchen. Wenn ein Service-Ereignis eintritt, können Sie die Lese- und Schreibvorgänge Ihrer Anwendung vorübergehend an die andere Region weiterleiten.

Bewährte Methoden für die Verwendung von MSK-Replikator

In diesem Abschnitt werden allgemeine bewährte Methoden und Implementierungsstrategien für die Verwendung von MSK-Replikator behandelt.

Themen

- Verwaltung des MSK-Replikator-Durchsatzes mithilfe von Kafka-Kontingenten
- Festlegen des Cluster-Aufbewahrungszeitraums

Verwaltung des MSK-Replikator-Durchsatzes mithilfe von Kafka-Kontingenten

Da MSK-Replikator als Verbraucher für Ihren Quell-Cluster fungiert, kann die Replikation dazu führen, dass andere Verbraucher im Quell-Cluster gedrosselt werden. Der Umfang der Drosselung hängt von der Lesekapazität Ihres Quell-Clusters und dem Datendurchsatz ab, den Sie replizieren. Wir empfehlen, dass Sie identische Kapazität für Ihre Quell- und Ziel-Cluster bereitstellen und den Replikationsdurchsatz bei der Berechnung der benötigten Kapazität berücksichtigen.

Sie können auch Kafka-Kontingente für den Replikator auf Ihren Quell- und Ziel-Clustern festlegen, um zu kontrollieren, wie viel Kapazität der MSK-Replikator nutzen kann. Ein Netzwerkbandbreiten-Kontingent wird empfohlen. Ein Netzwerkbandbreiten-Kontingent definiert einen Schwellenwert für

die Byterate, definiert als Bytes pro Sekunde, für einen oder mehrere Clients, die sich ein Kontingent teilen. Dieses Kontingent wird für jeden Broker individuell festgelegt.

Gehen Sie wie folgt vor, um ein Kontingent anzuwenden.

1. Rufen Sie die Bootstrap-Server-Zeichenfolge für den Quell-Cluster ab. Siehe Abrufen der Bootstrap-Broker für einen Amazon-MSK-Cluster.
2. Rufen Sie die vom MSK-Replikator verwendete Service-Ausführungsrolle (SER) ab. Dies ist die SER, die Sie für eine `CreateReplicator`-Anfrage verwendet haben. Sie können den SER auch aus der DescribeReplicator Antwort eines vorhandenen Replikators abrufen.
3. Führen Sie mit den Kafka-CLI-Tools den folgenden Befehl für den Quell-Cluster aus.

```
./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. Stellen Sie nach der Ausführung des obigen Befehls sicher, dass die `ReplicatorThroughput`-Metrik das von Ihnen festgelegte Kontingent nicht überschreitet.

Beachten Sie, dass, wenn Sie eine Service-Ausführungsrolle zwischen mehreren MSK-Replikatoren wiederverwenden, diese alle diesem Kontingent unterliegen. Wenn Sie separate Kontingente pro Replikator beibehalten möchten, verwenden Sie separate Service-Ausführungsrollen.

Weitere Informationen zur Verwendung der MSK-IAM-Authentifizierung mit Kontingenten finden Sie unter Multi-Tenancy-Apache-Kafka-Cluster in Amazon MSK mit IAM-Zugriffssteuerung und Kafka-Kontingente – Teil 1.

> **Warning**
>
> Die Einstellung einer extrem niedrigen consumer_byte_rate kann dazu führen, dass Ihr MSK-Replikator auf unerwartete Weise reagiert.

Festlegen des Cluster-Aufbewahrungszeitraums

Sie können den Aufbewahrungszeitraum für Protokolle für von MSK bereitgestellte Cluster und Serverless-Cluster festlegen. Der empfohlene Aufbewahrungszeitraum beträgt 7 Tage.

Weitere Informationen unter Änderungen der Cluster-Konfiguration oder MSK-Serverless-Cluster-Konfiguration.

Cluster-Status

Die folgende Tabelle zeigt die möglichen Status eines Clusters und beschreibt, was sie bedeuten. Außerdem wird beschrieben, welche Aktionen Sie ausführen können und welche nicht, wenn sich ein Cluster in einem dieser Status befindet. Um den Status eines Clusters herauszufinden, können Sie die AWS Management Console aufrufen. Sie können den Cluster auch mit dem Befehl describe-cluster-v2 oder der Operation DescribeClusterV2 beschreiben. Die Beschreibung eines Clusters beinhaltet seinen Status.

Cluster-Status	Bedeutung und mögliche Aktionen
ACTIVE	Sie können Daten produzieren und verbrauchen. Sie können auch die Amazon MSK-API und AWS CLI -Operationen auf dem Cluster ausführen.
WIRD ERSTELLT	Amazon MSK richtet den Cluster ein. Sie müssen warten, bis der Cluster den Status ACTIVE erreicht hat, bevor Sie ihn zur Erzeugung oder Nutzung von Daten oder zur Ausführung der Amazon MSK-API oder AWS CLI -Operationen verwenden können.
WIRD GELÖSCHT	Der Cluster wird gerade gelöscht. Sie können ihn nicht verwenden, um Daten zu erzeugen oder zu verbrauchen. Sie können auch keine Amazon MSK-API oder AWS CLI Operationen darauf ausführen.
FEHLGESCHLAGEN	Der Prozess zum Erstellen oder Löschen des Clusters ist fehlgeschlagen. Sie können den Cluster nicht zum Erstellen oder Verbrauchen von Daten verwenden. Sie können den Cluster löschen, aber keine Amazon MSK-API oder AWS CLI Aktualisierungsvorgänge darauf ausführen.

Cluster-Status	Bedeutung und mögliche Aktionen
HEALING	Amazon MSK führt einen internen Vorgang durch, z. B. den Austausch eines fehlerhaften Brokers. Beispielsweise reagiert der Broker möglicherweise nicht. Sie können den Cluster immer noch zum Erstellen oder Verbrauchen von Daten verwenden. Sie können jedoch keine Amazon MSK-API- oder AWS CLI Aktualisierungsvorgänge auf dem Cluster ausführen, bis er wieder in den Status ACTIVE zurückkehrt.
MAINTENANCE	Amazon MSK führt routinemäßige Wartungsvorgänge am Cluster durch. Zu diesen Wartungsvorgängen gehören auch Sicherheitspatches. Sie können den Cluster immer noch zum Erstellen oder Verbrauchen von Daten verwenden. Sie können jedoch keine Amazon MSK-API- oder AWS CLI Aktualisierungsvorgänge auf dem Cluster ausführen, bis er wieder in den Status ACTIVE zurückkehrt.
REBOOTING_BROKER	Amazon MSK startet einen Broker neu. Sie können den Cluster immer noch zum Erstellen oder Verbrauchen von Daten verwenden. Sie können jedoch keine Amazon MSK-API- oder AWS CLI Aktualisierungsvorgänge auf dem Cluster ausführen, bis er wieder in den Status ACTIVE zurückkehrt.

Cluster-Status	Bedeutung und mögliche Aktionen
WIRD AKTUALISIERT	Eine vom Benutzer initiierte Amazon MSK-API oder ein AWS CLI Vorgang aktualisiert den Cluster. Sie können den Cluster immer noch zum Erstellen oder Verbrauchen von Daten verwenden. Sie können jedoch keine weiteren Amazon MSK-API- oder AWS CLI Aktualisierungsvorgänge für den Cluster ausführen, bis er wieder in den Status ACTIVE zurückkehrt.

Sicherheit in Amazon Managed Streaming für Apache Kafka

Cloud-Sicherheit hat AWS höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Informationen zu den Compliance-Programmen, die für Amazon Managed Streaming für Apache Kafka gelten, finden Sie unter Im Rahmen des Compliance-Programms zugelassene Amazon-Web-Services.
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Dienst, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, einschließlich der Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation zeigt Ihnen, wie Sie das Modell der geteilten Verantwortung bei der Verwendung von Amazon MSK einsetzen können. Die folgenden Themen zeigen Ihnen, wie Sie Amazon MKS konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie lernen auch, wie Sie andere Amazon Web Services verwenden, die Sie bei der Überwachung und Sicherung Ihrer Amazon-MSK-Ressourcen unterstützen.

Themen

- Infrastruktursicherheit in Amazon Managed Streaming für Apache Kafka

Datenschutz in Amazon Managed Streaming für Apache Kafka

Das Modell der AWS gemeinsamen Verantwortung gilt für den Datenschutz in Amazon Managed Streaming for Apache Kafka. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der AWS Cloud alle Systeme laufen. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS -Modell der geteilten Verantwortung und in der DSGVO im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie SSL/TLS, um mit Ressourcen zu kommunizieren. AWS Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein. AWS CloudTrail
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-2.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit Amazon MSK oder anderen AWS-Services über die Konsole AWS CLI, API oder AWS SDKs arbeiten. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen

externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Themen

Amazon-MSK-Verschlüsselung

Amazon MSK bietet Datenverschlüsselungsoptionen, mit denen Sie strenge Anforderungen an die Datenverwaltung erfüllen können. Die Zertifikate, die Amazon MSK für die Verschlüsselung verwendet, müssen alle 13 Monate erneuert werden. Amazon MSK erneuert diese Zertifikate automatisch für alle Cluster. Der Status des Clusters wird auf `MAINTENANCE` festgelegt, wenn er die Operation „certificate-update“ startet. Es wird auf `ACTIVE` zurückgesetzt, wenn das Update abgeschlossen ist. Während sich ein Cluster im Status `MAINTENANCE` befindet, können Sie weiterhin Daten erstellen und verwenden, Sie können jedoch keine Aktualisierungsvorgänge für ihn ausführen.

Verschlüsselung im Ruhezustand

Amazon MSK wird in AWS Key Management Service (KMS) integriert, um transparente serverseitige Verschlüsselung zu ermöglichen. Amazon MQ verschlüsselt stets Ihre Daten im Ruhezustand. Wenn Sie einen MSK-Cluster erstellen, können Sie den AWS KMS key angeben, den Amazon MSK zur Verschlüsselung Ihrer Daten im Ruhezustand verwenden soll. Wenn Sie keinen KMS-Schlüssel angeben, erstellt Amazon MSK einen Von AWS verwalteter Schlüssel für Sie und verwendet ihn in Ihrem Namen. Weitere Informationen zum Verwenden von CMK-Schlüssel finden AWS KMS keysSie im AWS Key Management Service Entwicklerhandbuch.

Verschlüsselung während der Übertragung

Amazon MSK verwendet TLS 1.2. Daten werden standardmäßig während der Übertragung zwischen den Brokern Ihres MSK-Clusters verschlüsselt. Sie können diese Standardeinstellung beim Erstellen des Clusters außer Kraft setzen.

Für die Kommunikation zwischen Clients und Brokern müssen Sie eine der folgenden drei Einstellungen angeben:

- Nur Daten mit TLS-Verschlüsselung zulassen. Dies ist die Standardeinstellung.
- Sowohl Klartextdaten als auch Daten mit TLS-Verschlüsselung zulassen

- Nur Klartextdaten zulassen

Amazon MSK-Broker verwenden öffentliche AWS Certificate Manager Zertifikate. Daher vertraut jeder Vertrauensspeicher, der Amazon Trust Services vertraut, auch den Zertifikaten von Amazon-MSK-Brokern.

Während wir dringend empfehlen, die Verschlüsselung während der Übertragung zu aktivieren, kann dies zusätzliche CPU-Kosten und einige Millisekunden Latenz verursachen. Die meisten Anwendungsfälle reagieren jedoch nicht empfindlich auf diese Unterschiede, und das Ausmaß der Auswirkungen hängt von der Konfiguration Ihres Clusters, Ihrer Clients und Ihres Nutzungsprofils ab.

Wie kann ich mit der Verschlüsselung beginnen?

Beim Erstellen eines MSK-Clusters können Sie Verschlüsselungseinstellungen im JSON-Format angeben. Im Folgenden wird ein Beispiel gezeigt.

```
{
   "EncryptionAtRest": {
       "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/abcdabcd-1234-
abcd-1234-abcd123e8e8e"
    },
   "EncryptionInTransit": {
        "InCluster": true,
        "ClientBroker": "TLS"
    }
}
```

Für `DataVolumeKMSKeyId` können Sie einen vom Kunden verwalteten Schlüssel oder den Von AWS verwalteter Schlüssel für MSK in Ihrem Konto angeben (`alias/aws/kafka`). Wenn Sie nichts angeben`EncryptionAtRest`, verschlüsselt Amazon MSK Ihre ruhenden Daten trotzdem unter dem. Von AWS verwalteter Schlüssel Um festzustellen, welchen Schlüssel Ihr Cluster verwendet, senden Sie eine `GET`-Anforderung oder rufen Sie den `DescribeCluster`-API-Vorgang auf.

Für `EncryptionInTransit` ist der Standardwert von `InCluster` auf Wahr festgelegt, aber Sie können ihn auf Falsch setzen, wenn Sie Ihre Daten bei der Übergabe zwischen Brokern nicht von Amazon MSK verschlüsseln lassen möchten.

Um den Verschlüsselungsmodus für die Übertragung von Daten zwischen Clients und Brokern anzugeben, legen Sie `ClientBroker` auf einen der drei Werte folgenden fest: `TLS`, `TLS_PLAINTEXT`, oder `PLAINTEXT`.

So legen Sie die Verschlüsselungseinstellungen beim Erstellen eines Clusters fest

1. Speichern Sie den Inhalt des vorherigen Beispiels in einer Datei und geben Sie der Datei einen beliebigen Namen. Nennen Sie sie beispielsweise „`encryption-settings.json`“.
2. Führen Sie den `create-cluster`-Befehl aus, und weisen Sie mithilfe der `encryption-info`-Option auf die Datei, in der Sie Ihr Konfigurations-JSON gespeichert haben. Im Folgenden wird ein Beispiel gezeigt. Ersetzen Sie *`{YOUR MSK VERSION}`* durch eine Version, die der Apache-Kafka-Client-Version entspricht. Weitere Informationen zum Auffinden der MSK-Cluster-Version finden Sie unter To find the version of your MSK cluster. Beachten Sie, dass die Verwendung einer Apache-Kafka-Client-Version, die nicht mit Ihrer MSK-Cluster-Version identisch ist, zu Beschädigung, Verlust und Ausfallzeiten von Apache-Kafka-Daten führen kann.

```
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
```

   Im Folgenden finden Sie ein Beispiel für eine erfolgreiche Antwort nach der Ausführung dieses Befehls.

```
{
    "ClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/SecondTLSTest/
abcdabcd-1234-abcd-1234-abcd123e8e8e",
    "ClusterName": "ExampleClusterName",
    "State": "CREATING"
}
```

So testen Sie die TLS-Verschlüsselung

1. Erstellen Sie einen Client-Computer entsprechend der Anweisungen in the section called “Schritt 3: Einen Client-Computer erstellen”.
2. Installieren Sie Apache Kafka auf dem Client-Computer.
3. In diesem Beispiel verwenden wir den JVM-Vertrauensspeicher, um mit dem MSK-Cluster zu kommunizieren. Erstellen Sie dazu zunächst einen Ordner mit dem Namen `/tmp` auf dem Client-Computer. Wechseln Sie dann zum Ordner „`bin`“ der Apache Kafka-Installation und führen Sie den folgenden Befehl aus. (Ihr JVM-Pfad kann sich unterscheiden.)

```
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. Wenn Sie sich noch im `bin`-Ordner der Apache Kafka-Installation auf dem Client-Computer befinden, erstellen Sie eine Textdatei `client.properties` mit dem folgenden Inhalt.

```
security.protocol=SSL
ssl.truststore.location=/tmp/kafka.client.truststore.jks
```

5. Führen Sie den folgenden Befehl auf einem Computer aus, auf dem das AWS CLI installiert ist, und ersetzen Sie *`ClusterArn`* durch den ARN Ihres Clusters.

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

Ein erfolgreiches Ergebnis sieht wie folgt aus. Speichern Sie dieses Ergebnis, da Sie es für den nächsten Schritt benötigen.

```
{
    "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. Führen Sie den folgenden Befehl aus und *`BootstrapBrokerStringTls`*ersetzen Sie ihn durch einen der Broker-Endpunkte, die Sie im vorherigen Schritt erhalten haben.

```
<path-to-your-kafka-installation>/bin/kafka-console-producer.sh --broker-
list BootstrapBrokerStringTls --producer.config client.properties --topic
 TLSTestTopic
```

7. Öffnen Sie ein neues Befehlsfenster und stellen Sie eine Verbindung zu demselben Client-Computer her. Führen Sie dann den folgenden Befehl aus, um einen Konsolenverbraucher zu erstellen.

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBrokerStringTls --consumer.config client.properties --topic
 TLSTestTopic
```

8. Geben Sie im Produzent-Fenster eine Textnachricht gefolgt von einem Zeilenumbruch ein, und suchen Sie im Verbraucher-Fenster nach derselben Nachricht. Amazon MSK hat diese Nachricht während der Übertragung verschlüsselt.

Weitere Informationen zum Konfigurieren von Apache Kafka-Clients für die Arbeit mit verschlüsselten Daten finden Sie unter Konfigurieren von Kafka-Clients.

Authentifizierung und Autorisierung für Amazon-MSK-APIs

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren steuern, wer authentifiziert (angemeldet) und autorisiert (im Besitz von Berechtigungen) ist, Amazon-MSK-Ressourcen zu nutzen. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Auf dieser Seite wird beschrieben, wie Sie IAM verwenden können, um zu steuern, wer Amazon-MSK-Vorgänge in Ihrem Cluster ausführen kann. Informationen darüber, wie Sie steuern können, wer Apache-Kafka-Vorgänge auf Ihrem Cluster ausführen kann, finden Sie unter the section called “Authentifizierung und Autorisierung für Apache-Kafka-APIs”.

Themen

- Funktionsweise von Amazon MKS mit IAM
- Beispiele für identitätsbasierte Amazon-MSK-Richtlinien
- Verwendung von serviceverknüpften Rollen für Amazon MSK
- AWS verwaltete Richtlinien für Amazon MSK
- Fehlerbehebung für Amazon-MSK-Identität und -Zugriff

Funktionsweise von Amazon MKS mit IAM

Bevor Sie mit IAM den Zugriff auf Amazon MSK verwalten können, sollten Sie sich darüber informieren, welche IAM-Funktionen Sie mit Amazon MSK verwenden können. Einen allgemeinen Überblick darüber, wie Amazon MSK und andere AWS Services mit IAM zusammenarbeiten, finden Sie unter AWS Services That Work with IAM im IAM-Benutzerhandbuch.

Themen

- Identitätsbasierte Amazon-MSK-Richtlinien

- Ressourcenbasierte Amazon-MSK-Richtlinien
- AWS verwaltete Richtlinien
- Autorisierung basierend auf Amazon-MSK-Tags
- Amazon-MSK-IAM-Rollen

Identitätsbasierte Amazon-MSK-Richtlinien

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Amazon MSK unterstützt bestimmte Aktionen, Ressourcen und Bedingungsschlüssel. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Aktionen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Richtlinienaktionen in Amazon MSK verwenden das folgende Präfix vor der Aktion: `kafka:`. Wenn Sie beispielsweise einem Benutzer die Berechtigung erteilen möchten, einen MSK-Cluster mit dem Amazon-MSK-API-Vorgang `DescribeCluster` zu beschreiben, nehmen Sie die Aktion `kafka:DescribeCluster` in die Richtlinie auf. Richtlinienanweisungen müssen entweder ein `Action`- oder ein `NotAction`-Element enthalten. Amazon MSK definiert einen eigenen Satz von Aktionen, die Aufgaben beschreiben, die Sie mit diesem Service durchführen können.

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie wie folgt durch Kommata:

```
"Action": ["kafka:action1", "kafka:action2"]
```

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

```
"Action": "kafka:Describe*"
```

Eine Liste der Amazon-MSK-Aktionen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Managed Streaming für Apache Kafka im IAM-Benutzerhandbuch.

Ressourcen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein – `Resource`oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Die Amazon-MSK-Instance-Ressource hat den folgenden ARN:

```
arn:${Partition}:kafka:${Region}:${Account}:cluster/${ClusterName}/${UUID}
```

Weitere Informationen zum Format von ARNs finden Sie unter Amazon Resource Names (ARNs) und AWS Service Namespaces.

Wenn Sie beispielsweise die `CustomerMessages`-Instance in Ihrer Anweisung angeben möchten, verwenden Sie den folgenden ARN:

```
"Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/CustomerMessages/abcd1234-abcd-dcba-4321-a1b2abcd9f9f-2"
```

Um alle Instances anzugeben, die zu einem bestimmten Konto gehören, verwenden Sie den Platzhalter (*):

```
"Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/*"
```

Einige Amazon-MKS-Aktionen, z. B. das Erstellen von Ressourcen, können für bestimmte Ressourcen nicht ausgeführt werden. In diesen Fällen müssen Sie den Platzhalter (*) verwenden.

```
"Resource": "*"
```

Um mehrere Ressourcen in einer einzigen Anweisung anzugeben, trennen Sie die ARNs durch Kommata voneinander.

```
"Resource": ["resource1", "resource2"]
```

Eine Liste der Amazon-MSK-Ressourcen-Typen und ihrer ARNs finden Sie unter Von Amazon Managed Streaming für Apache Kafka definierte Ressourcen im IAM-Benutzerhandbuch. Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter Von Amazon Managed Streaming für Apache Kafka definierte Aktionen.

Bedingungsschlüssel

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen `AND`-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen `OR` Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

Amazon MSK definiert seinen eigenen Satz von Bedingungsschlüsseln und unterstützt auch die Verwendung einiger globaler Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter AWS Globale Bedingungskontextschlüssel im IAM-Benutzerhandbuch.

Eine Liste der Amazon-MSK-Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für Amazon Managed Streaming für Apache Kafka im IAM-Benutzerhandbuch. Informationen dazu, mit welchen Aktionen und Ressourcen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von Amazon Managed Streaming für Apache Kafka definierte Aktionen.

Beispiele

Beispiele für identitätsbasierte Amazon-MSK-Richtlinien finden Sie unter Beispiele für identitätsbasierte Amazon-MSK-Richtlinien.

Ressourcenbasierte Amazon-MSK-Richtlinien

Amazon MSK unterstützt eine Cluster-Richtlinie (auch als ressourcenbasierte Richtlinie bezeichnet) zur Verwendung mit Amazon-MSK-Clustern. Sie können eine Cluster-Richtlinie verwenden, um zu definieren, welche IAM-Prinzipale über kontoübergreifende Berechtigungen zum Einrichten einer privaten Konnektivität mit Ihrem Amazon-MSK-Cluster verfügen. Bei Verwendung mit der IAM-Client-Authentifizierung können Sie die Cluster-Richtlinie auch verwenden, um die Kafka-Datenebenen-Berechtigungen für die verbindenden Clients detailliert zu definieren.

Ein Beispiel für die Konfiguration einer Cluster-Richtlinie finden Sie unter Schritt 2: Eine Cluster-Richtlinie an den MSK-Cluster anhängen.

AWS verwaltete Richtlinien

Autorisierung basierend auf Amazon-MSK-Tags

Sie können Amazon-MSK-Clustern Tags anhängen. Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `kafka:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden. Weitere Informationen über das Markieren von Amazon-MSK-Ressourcen finden Sie unter the section called “Markieren eines Clusters”.

Eine exemplarische identitätsbasierte Richtlinie zur Beschränkung des Zugriffs auf einen Cluster basierend auf den Tags dieses Clusters finden Sie unter Zugreifen auf Amazon-MSK-Cluster anhand von Tags.

Amazon-MSK-IAM-Rollen

Eine IAM-Rolle ist eine Entität in Ihrem Amazon–Web-Services-Konto mit spezifischen Berechtigungen.

Verwenden temporärer Anmeldeinformationen mit Amazon MSK

Sie können temporäre Anmeldeinformationen verwenden, um sich über einen Verbund anzumelden, eine IAM-Rolle anzunehmen oder eine kontenübergreifende Rolle anzunehmen. Sie erhalten temporäre Sicherheitsanmeldedaten, indem Sie AWS STS API-Operationen wie AssumeRoleoder GetFederationToken aufrufen.

Amazon MSK unterstützt die Verwendung temporärer Anmeldeinformationen.

Service-verknüpfte Rollen

Serviceverknüpfte Rollen erlauben Amazon Web Services den Zugriff auf Ressourcen in anderen Services, um eine Aktion in Ihrem Auftrag auszuführen. Serviceverknüpfte Rollen werden in Ihrem IAM-Konto angezeigt und gehören zum Service. Ein -Administrator kann die Berechtigungen für serviceverknüpfte Rollen anzeigen, aber nicht bearbeiten.

Amazon MSK unterstützt serviceverknüpfte Rollen. Weitere Informationen zum Erstellen oder Verwalten von serviceverknüpften Amazon-MSK-Rollen finden Sie unter the section called “Service-verknüpfte Rollen”.

Beispiele für identitätsbasierte Amazon-MSK-Richtlinien

Standardmäßig haben IAM-Benutzer und -Rollen keine Berechtigung zum Ausführen von Amazon-MSK-API-Aktionen. Ein Administrator muss IAM-Richtlinien erstellen, die Benutzern und Rollen

die Berechtigung zum Ausführen bestimmter API-Operationen für die angegebenen Ressourcen gewähren, die diese benötigen. Der Administrator muss diese Richtlinien anschließend den IAM-Benutzern oder -Gruppen anfügen, die diese Berechtigungen benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von Richtlinien auf der JSON-Registerkarte im IAM-Benutzerhandbuch.

Themen

- Bewährte Methoden für Richtlinien
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer
- Zugriff auf einen Amazon-MSK-Cluster
- Zugreifen auf Amazon-MSK-Cluster anhand von Tags

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien können festlegen, ob jemand Amazon-MSK-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter AWS -verwaltete Richtlinien oder AWS -verwaltete Richtlinien für Auftrags-Funktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.

- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung zum IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Konfigurieren eines MFA-geschützten API-Zugriffs im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API oder. 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": "*"
        }
    ]
}
```

Zugriff auf einen Amazon-MSK-Cluster

In diesem Beispiel möchten Sie einem IAM-Benutzer in Ihrem Amazon-Web-Services-Konto den Zugriff auf einen Ihrer Cluster gewähren, `purchaseQueriesCluster`. Diese Richtlinie ermöglicht es dem Benutzer, den Cluster zu beschreiben, seine Bootstrap-Broker abrufen, seine Broker-Knoten auflisten und ihn zu aktualisieren.

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

Zugreifen auf Amazon-MSK-Cluster anhand von Tags

Sie können Bedingungen in Ihrer identitätsbasierten Richtlinie verwenden, um den Zugriff auf Amazon-MSK-Ressourcen anhand von Tags zu steuern. In diesem Beispiel wird dargestellt, wie Sie eine Richtlinie erstellen können, mit der Benutzer den Cluster beschreiben, seine Bootstrap-Broker abrufen, seine Broker-Knoten auflisten, ihn aktualisieren und löschen können. Die Berechtigung wird jedoch nur gewährt, wenn der Wert des Cluster-Tags „`Owner`“ dem Benutzernamen entspricht.

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

Sie können diese Richtlinie den IAM-Benutzern in Ihrem Konto anfügen. Wenn ein Benutzer mit dem Namen `richard-roe` versucht, einen MSK-Cluster zu aktualisieren, muss der Cluster mit dem Tag `Owner=richard-roe` oder `owner=richard-roe` markiert sein. Andernfalls wird der

Zugriff abgelehnt. Der Tag-Schlüssel `0wner` der Bedingung stimmt sowohl mit `0wner` als auch mit `owner` überein, da die Namen von Bedingungsschlüsseln nicht zwischen Groß- und Kleinschreibung unterscheiden. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.

Verwendung von serviceverknüpften Rollen für Amazon MSK

Amazon MSK verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit Amazon MSK verknüpft ist. Servicebezogene Rollen sind von Amazon MSK vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

Eine serviceverknüpfte Rolle macht die Einrichtung von Amazon MSK einfacher, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Amazon MSK definiert die Berechtigungen seiner serviceverknüpften Rollen. Sofern nicht anders definiert, kann nur Amazon MSK seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter Amazon Web Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Themen

- Serviceverknüpfte Rollenberechtigungen für Amazon MSK
- Erstellen einer serviceverknüpften Rolle für Amazon MSK
- Bearbeiten einer serviceverknüpften Rolle für Amazon MSK
- Unterstützte Regionen für Amazon MSK serviceverknüpfte Rollen

Serviceverknüpfte Rollenberechtigungen für Amazon MSK

Amazon MSK verwendet die serviceverknüpfte Rolle namens AWSServiceRoleForKafka. Amazon MSK verwendet diese Rolle für den Zugriff auf Ihre Ressourcen und für die Ausführung von Vorgängen wie:

- `*NetworkInterface` – Netzwerkschnittstellen im Kundenkonto erstellen und verwalten, die Cluster-Broker für Clients in der Kunden-VPC zugänglich machen.

- `*VpcEndpoints`— VPC-Endpunkte im Kundenkonto verwalten, die Cluster-Broker für Kunden in der Kunden-VPC zugänglich machen, die sie verwenden. AWS PrivateLink Amazon MSK verwendet Berechtigungen für `DescribeVpcEndpoints`, `ModifyVpcEndpoint` und `DeleteVpcEndpoints`.
- `secretsmanager`— Kundenanmeldedaten verwalten mit. AWS Secrets Manager
- `GetCertificateAuthorityCertificate` – Das Zertifikat für Ihre private Zertifizierungsstelle abrufen.

Diese verwaltete Richtlinie ist mit der folgenden serviceverknüpften Rolle verbunden: `KafkaServiceRolePolicy`. Aktualisierungen dieser Richtlinie finden Sie unter KafkaServiceRolePolicy.

Die serviceverknüpfte Rolle AWSServiceRoleForKafka vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `kafka.amazonaws.com`

Die Rollenberechtigungsrichtlinie erlaubt es Amazon MSK, die folgenden Aktionen für Ressourcen durchzuführen.

```
{
 "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_*"
     }
    }
   }
  ]
}
```

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter serviceverknüpfte Rollenberechtigungen im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für Amazon MSK

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie einen Amazon MSK-Cluster in der AWS Management Console, der oder der AWS API erstellen AWS CLI, erstellt Amazon MSK die serviceverknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie einen Amazon-MSK-Cluster erstellen, erstellt Amazon MSK wieder die servicevereknüpfte Rolle für Sie.

Bearbeiten einer serviceverknüpften Rolle für Amazon MSK

Amazon MSK verhindert die Bearbeitung der serviceverknüpften Rolle AWSServiceRoleForKafka. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für Amazon MSK serviceverknüpfte Rollen

Amazon MSK unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter AWS -Regionen und Endpunkte.

AWS verwaltete Richtlinien für Amazon MSK

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet AWS wird. AWS Verwaltete Richtlinien dienen dazu, Berechtigungen für viele gängige Anwendungsfälle bereitzustellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie allen AWS Kunden zur Verfügung stehen. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie kundenverwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinie: AmazonMSK FullAccess

Diese Richtlinie gewährt Administratorberechtigungen, die einem Prinzipal vollen Zugriff auf alle Amazon-MSK-Aktionen erlauben. Die Berechtigungen in dieser Richtlinie sind wie folgt gruppiert:

- Die Amazon-MSK-Berechtigungen erlauben alle Amazon-MSK-Aktionen.
- **`Amazon EC2`**Berechtigungen — In dieser Richtlinie sind sie erforderlich, um die übergebenen Ressourcen in einer API-Anfrage zu validieren. Dadurch soll sichergestellt werden, dass Amazon MSK die Ressourcen erfolgreich mit einem Cluster nutzen kann. Die übrigen Amazon EC2 EC2-Berechtigungen in dieser Richtlinie ermöglichen es Amazon MSK, AWS Ressourcen zu erstellen, die erforderlich sind, damit Sie eine Verbindung zu Ihren Clustern herstellen können.
- **`AWS KMS`**Berechtigungen — werden bei API-Aufrufen verwendet, um die übergebenen Ressourcen in einer Anfrage zu validieren. Sie sind erforderlich, damit Amazon MSK den übergebenen Schlüssel mit dem Amazon-MSK-Cluster verwenden kann.
- **`CloudWatch Logs, Amazon S3, and Amazon Data Firehose`**Berechtigungen — sind erforderlich, damit Amazon MSK sicherstellen kann, dass die Protokollzustellungsziele erreichbar sind und dass sie für die Verwendung von Broker-Protokollen gültig sind.
- **`IAM`**Berechtigungen — sind erforderlich, damit Amazon MSK eine serviceverknüpfte Rolle in Ihrem Konto erstellen und eine Rolle zur Ausführung von Dienstleistungen an Amazon MSK übergeben kann.

```
{
 "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 verwaltete Richtlinie: AmazonMSK Access ReadOnly

Diese Richtlinie gewährt schreibgeschützte Berechtigungen, die es Benutzern erlauben, Informationen in Amazon MSK anzuzeigen. Prinzipale, denen diese Richtlinie angefügt ist, können keine Aktualisierungen vornehmen oder bestehende Ressourcen löschen. Sie können auch keine neuen Amazon-MSK-Ressourcen erstellen. Prinzipale mit diesen Berechtigungen können beispielsweise die Liste der Cluster und Konfigurationen, die mit ihrem Konto verknüpft sind, einsehen, aber nicht die Konfiguration oder Einstellungen von Clustern ändern. Die Berechtigungen in dieser Richtlinie sind wie folgt gruppiert:

- **`Amazon MSK`**Berechtigungen — ermöglichen es Ihnen, Amazon MSK-Ressourcen aufzulisten, zu beschreiben und Informationen über sie abzurufen.
- **`Amazon EC2`**Berechtigungen — werden verwendet, um die Amazon VPC, Subnetze, Sicherheitsgruppen und ENIs zu beschreiben, die einem Cluster zugeordnet sind.
- **`AWS KMS`**Permission — wird verwendet, um den Schlüssel zu beschreiben, der dem Cluster zugeordnet ist.

```
{
    "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 verwaltete Richtlinie: KafkaServiceRolePolicy

Sie können keine Verbindungen KafkaServiceRolePolicy zu Ihren IAM-Entitäten herstellen. Diese Richtlinie ist mit einer servicegebundenen Rolle verknüpft, die es Amazon MSK ermöglicht, Aktionen wie die Verwaltung von VPC-Endpunkten (Konnektoren) auf MSK-Clustern, die Verwaltung von Netzwerkschnittstellen und die Verwaltung von Cluster-Anmeldeinformationen mit AWS Secrets Manager durchzuführen. Weitere Informationen finden Sie unter the section called “Service-verknüpfte Rollen”.

AWS verwaltete Richtlinie: AWSMSKReplicatorExecutionRole

Die `AWSMSKReplicatorExecutionRole` Richtlinie gewährt dem Amazon MSK-Replikator die Erlaubnis, Daten zwischen MSK-Clustern zu replizieren. Die Berechtigungen in dieser Richtlinie sind wie folgt gruppiert:

- **`cluster`**— Erteilt dem Amazon MSK Replicator die Berechtigung, mithilfe der IAM-Authentifizierung eine Verbindung zum Cluster herzustellen. Erteilt außerdem Berechtigungen zur Beschreibung und Änderung des Clusters.
- **`topic`**— Erteilt dem Amazon MSK Replicator Berechtigungen zum Beschreiben, Erstellen und Ändern eines Themas sowie zum Ändern der dynamischen Konfiguration des Themas.

- **`consumer group`**— Erteilt dem Amazon MSK Replicator Berechtigungen zum Beschreiben und Ändern von Nutzergruppen, zum Lesen und Schreiben von Daten aus einem MSK-Cluster und zum Löschen interner Themen, die vom Replikator erstellt wurden.

```
{
 "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 aktualisiert AWS verwaltete Richtlinien

Sehen Sie sich Details zu Aktualisierungen der AWS verwalteten Richtlinien für Amazon MSK an, seit dieser Service begonnen hat, diese Änderungen zu verfolgen.

Änderung	Beschreibung	Datum
WriteDataIdempotently Berechtigung hinzugefügt zu AWSMSKReplicatorExecutionRole — Aktualisierung einer bestehenden Richtlinie	Amazon MSK hat der AWSMSKReplicatorExecutionRole Richtlinie die WriteDataIdempotently Erlaubnis zur Unterstützung der Datenreplikation zwischen MSK-Clustern hinzugefügt.	12. März 2024
AWSMSKReplicatorExecutionRole – Neue Richtlinie.	Amazon MSK hat eine AWSMSKReplicatorExecutionRole Richtlinie zur Unterstützung von Amazon MSK Replicator hinzugefügt.	4. Dezember 2023

Änderung	Beschreibung	Datum
AmazonMSK FullAccess — Aktualisierung einer bestehenden Richtlinie	Amazon MSK hat Berechtigungen zur Unterstützung von Amazon MSK Replicator hinzugefügt.	28. September 2023
KafkaServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon MSK hat Berechtigungen zur Unterstützung privater Multi-VPC-Konnektivität hinzugefügt.	08. März 2023
AmazonMSK FullAccess — Aktualisierung einer bestehenden Richtlinie	Amazon MSK hat neue Amazon-EC2-Berechtigungen hinzugefügt, um die Verbindung zu einem Cluster zu ermöglichen.	30. November 2021
AmazonMSK FullAccess — Aktualisierung einer bestehenden Richtlinie	Amazon MSK hat eine neue Berechtigung hinzugefügt, mit der Amazon-EC2-Routing-Tabellen beschrieben werden können.	19. November 2021
Amazon MSK hat mit der Nachverfolgung von Änderungen begonnen	Amazon MSK hat damit begonnen, Änderungen an seinen AWS verwalteten Richtlinien nachzuverfolgen.	19. November 2021

Fehlerbehebung für Amazon-MSK-Identität und -Zugriff

Diagnostizieren und beheben Sie mithilfe der folgenden Informationen gängige Probleme, die bei der Verwendung von Amazon MSK und IAM auftreten können.

Themen

- Ich bin nicht autorisiert, eine Aktion in Amazon MSK auszuführen

Ich bin nicht autorisiert, eine Aktion in Amazon MSK auszuführen

Wenn Ihnen AWS Management Console mitgeteilt wird, dass Sie nicht berechtigt sind, eine Aktion durchzuführen, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson`-IAM-Benutzer versucht, die Konsole zum Löschen eines Clusters zu verwenden, jedoch nicht über `kafka:DeleteCluster`-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 kafka:DeleteCluster on resource: purchaseQueriesCluster
```

In diesem Fall bittet Mateo seinen Administrator um die Aktualisierung seiner Richtlinien, um unter Verwendung der Aktion `purchaseQueriesCluster` auf die Ressource `kafka:DeleteCluster` zugreifen zu können.

Authentifizierung und Autorisierung für Apache-Kafka-APIs

Sie können IAM verwenden, um Clients zu authentifizieren und Apache-Kafka-Aktionen zu erlauben oder zu verweigern. Alternativ können Sie TLS oder SASL/SCRAM verwenden, um Clients zu authentifizieren, und Apache-Kafka-ACLs, um Aktionen zu erlauben oder zu verweigern.

Informationen darüber, wie Sie steuern können, wer Amazon-MSK-Vorgänge auf Ihrem Cluster ausführen kann, finden Sie unter the section called “Authentifizierung und Autorisierung für Amazon-MSK-APIs”.

Themen

- IAM-Zugriffssteuerung
- Gegenseitige TLS-Authentifizierung
- Authentifizierung der Anmeldedaten mit AWS Secrets Manager
- Apache Kafka ACLs

IAM-Zugriffssteuerung

IAM-Zugriffssteuerung für Amazon MSK ermöglicht es Ihnen, sowohl die Authentifizierung als auch die Autorisierung für Ihren MSK-Cluster zu verwalten. Dies macht die Verwendung eines

Mechanismus für die Authentifizierung und einen anderen für die Autorisierung überflüssig. Wenn ein Client beispielsweise versucht, in Ihren Cluster zu schreiben, prüft Amazon MSK mithilfe von IAM, ob es sich bei diesem Client um eine authentifizierte Identität handelt und ob er berechtigt ist, für Ihren Cluster zu produzieren. Die IAM-Zugriffskontrolle funktioniert für Java- und Nicht-Java-Clients, einschließlich Kafka-Clients, die in Python JavaScript, Go und.NET geschrieben sind.

Amazon MSK protokolliert Zugriffsereignisse, sodass Sie sie prüfen können. Weitere Informationen finden Sie unter the section called “CloudTrail Ereignisse”.

Um die IAM-Zugriffssteuerung zu ermöglichen, nimmt Amazon MSK geringfügige Änderungen am Apache-Kafka-Quellcode vor. Diese Änderungen werden keinen spürbaren Unterschied in Ihrem Apache-Kafka-Erlebnis bewirken.

Important

Die IAM-Zugriffskontrolle gilt nicht für Apache-Knoten. ZooKeeper Weitere Informationen zum Steuern des Zugriffs auf diese Knoten finden Sie unter the section called “Kontrolle des Zugriffs auf Apache ZooKeeper”.

Important

Die Apache-Kafka-Einstellung `allow.everyone.if.no.acl.found` hat keine Auswirkung, wenn Ihr Cluster die IAM-Zugriffssteuerung verwendet.

Important

Sie können Apache-Kafka-ACL-APIs für einen MSK-Cluster aufrufen, der IAM-Zugriffssteuerung verwendet. Apache Kafka-ACLs haben jedoch keinen Einfluss auf die Autorisierung für IAM-Rollen. Sie müssen IAM-Richtlinien verwenden, um den Zugriff für IAM-Rollen zu steuern.

So funktioniert die IAM-Zugriffssteuerung für Amazon MSK

Um die IAM-Zugriffssteuerung für Amazon MSK zu verwenden, führen Sie die folgenden Schritte aus, die im Rest dieses Abschnitts ausführlich beschrieben werden.

- the section called “Erstellen Sie einen Cluster, der IAM-Zugriffssteuerung verwendet”
- the section called “Konfiguration von Clients für die IAM-Zugriffssteuerung”
- the section called “Autorisierungsrichtlinien erstellen”
- the section called “Bootstrap-Broker für IAM-Zugriffssteuerung abrufen”

Erstellen Sie einen Cluster, der IAM-Zugriffssteuerung verwendet

In diesem Abschnitt wird erklärt, wie Sie die AWS Management Console, die API oder die verwenden können, um einen Cluster AWS CLI zu erstellen, der die IAM-Zugriffskontrolle verwendet. Informationen zum Aktivieren der IAM-Zugriffssteuerung für einen vorhandenen Cluster finden Sie unter the section called “Aktualisieren der Sicherheit”.

Verwenden Sie den AWS Management Console , um einen Cluster zu erstellen, der die IAM-Zugriffskontrolle verwendet

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
2. Wählen Sie Cluster erstellen.
3. Wählen Sie Cluster mit benutzerdefinierten Einstellungen erstellen.
4. Wählen Sie im Abschnitt Authentifizierung die Option IAM-Zugriffssteuerung aus.
5. Führen Sie den Rest des Workflows zum Erstellen eines Clusters aus.

Verwenden Sie die API oder die AWS CLI , um einen Cluster zu erstellen, der die IAM-Zugriffskontrolle verwendet

- Um einen Cluster mit aktivierter IAM-Zugriffskontrolle zu erstellen, verwenden Sie die CreateClusterAPI oder den CLI-Befehl create-cluster und übergeben Sie den folgenden JSON-Code für den `ClientAuthentication` Parameter:. `"ClientAuthentication": { "Sasl": { "Iam": { "Enabled": true } }`

Konfiguration von Clients für die IAM-Zugriffssteuerung

Damit Clients mit einem MSK-Cluster kommunizieren können, der die IAM-Zugriffskontrolle verwendet, können Sie einen der folgenden Mechanismen verwenden:

- Konfiguration von anderen Clients als Java-Clients mithilfe des SASL_OAUTHBEARER-Mechanismus

- Konfiguration von Java-Clients mithilfe des SASL_OAUTHBEARER- oder AWS_MSK_IAM-Mechanismus

Verwenden des SASL_OAUTHBEARER-Mechanismus zur Konfiguration von IAM

1. Bearbeiten Sie Ihre client.properties-Konfigurationsdatei und nehmen Sie dafür die hervorgehobene Syntax im Python-Kafka-Beispielclient unten als Leitfaden. Konfigurationsänderungen sind in anderen Sprachen ähnlich.

```
#!/usr/bin/python3from kafka import KafkaProducer
from kafka.errors import KafkaError
import socket
import time
from aws_msk_iam_sasl_signer import MSKAuthTokenProvider

class MSKTokenProvider():
    def token(self):
        token, _ = MSKAuthTokenProvider.generate_auth_token('<my aws region>')
        return token

tp = MSKTokenProvider()

producer = KafkaProducer(
    bootstrap_servers='<my bootstrap string>',
    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. Laden Sie die Hilfsbibliothek für die von Ihnen gewählte Konfigurationssprache herunter und folgen Sie den Anweisungen im Abschnitt Erste Schritte auf der Homepage dieser Sprachbibliothek.
   - 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-started
   - Go: https://github.com/aws/aws-msk-iam-sasl-signer-go#getting-started
   - .NET: https://github.com/aws/aws-msk-iam-sasl-signer-net#getting-started
   - JAVA: Die SASL_OAUTHBEARER-Unterstützung für Java ist über die JAR-Datei `aws-msk-iam-auth`verfügbar

Verwenden des benutzerdefinierten MSK-Mechanismus AWS_MSK_IAM zur Konfiguration von IAM

1. Fügen Sie der Datei `client.properties` Folgendes hinzu. Ersetzen Sie *`<PATH_TO_TRUST_STORE_FILE>`* durch den vollqualifizierten Pfad zur Vertrauensspeicher-Datei auf dem Client.

   > **Note**
   >
   > Wenn Sie ein bestimmtes Zertifikat nicht verwenden möchten, können Sie `ssl.truststore.location=`*`<PATH_TO_TRUST_STORE_FILE>`* aus Ihrer `client.properties`-Datei entfernen. Wenn Sie keinen Wert für `ssl.truststore.location` angeben, verwendet der Java-Prozess das Standardzertifikat.

   ```
   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
   ```

   Um ein benanntes Profil zu verwenden, das Sie für AWS Anmeldeinformationen erstellt haben, nehmen Sie es `awsProfileName="`*`your profile name`*`";` in Ihre Client-Konfigurationsdatei auf. Informationen zu benannten Profilen finden Sie in der AWS CLI Dokumentation unter Benannte Profile.

2. Laden Sie die neueste stabile aws-msk-iam-auth-JAR-Datei herunter und platzieren Sie sie im Klassenpfad. Wenn Sie Maven verwenden, fügen Sie die folgende Abhängigkeit hinzu und passen Sie die Versionsnummer nach Bedarf an:

```
<dependency>
    <groupId>software.amazon.msk</groupId>
    <artifactId>aws-msk-iam-auth</artifactId>
    <version>1.0.0</version>
</dependency>
```

Das Amazon-MSK-Client-Plugin ist unter der Apache-2.0-Lizenz als Open-Source verfügbar.

Autorisierungsrichtlinien erstellen

Fügen Sie eine Autorisierungsrichtlinie an die IAM-Rolle an, die dem Client entspricht. In einer Autorisierungsrichtlinie geben Sie an, welche Aktionen für die Rolle erlaubt oder verweigert werden sollen. Wenn sich Ihr Client auf einer Amazon-EC2-Instance befindet, ordnen Sie die Autorisierungsrichtlinie der IAM-Rolle für diese Amazon-EC2-Instance zu. Alternativ können Sie Ihren Client so konfigurieren, dass er ein benanntes Profil verwendet, und dann die Autorisierungsrichtlinie der Rolle für dieses benannte Profil zuordnen. the section called “Konfiguration von Clients für die IAM-Zugriffssteuerung” beschreibt, wie ein Client für die Verwendung eines benannten Profils konfiguriert wird.

Informationen zum Erstellen einer IAM-Richtlinie finden Sie unter Erstellen von IAM-Richtlinien.

Im Folgenden finden Sie ein Beispiel für eine Autorisierungsrichtlinie für einen Cluster mit dem Namen MyTestCluster. Informationen zur Semantik der `Action`- und `Resource`-Elemente finden Sie unter the section called “Semantik von Aktionen und Ressourcen”.

> **Important**
>
> Änderungen, die Sie an einer IAM-Richtlinie vornehmen, werden in den IAM-APIs und der AWS CLI sofort wiedergegeben. Es kann jedoch einige Zeit dauern, bis die Änderung der Richtlinie wirksam wird. In den meisten Fällen werden Richtlinien-Änderungen in weniger als einer Minute wirksam. Netzwerkbedingungen können die Verzögerung manchmal erhöhen.

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:AlterCluster",
                "kafka-cluster:DescribeCluster"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:0123456789012: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-1:0123456789012:topic/MyTestCluster/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeGroup"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:0123456789012:group/MyTestCluster/*"
            ]
        }
    ]
}
```

Informationen zum Erstellen einer Richtlinie mit Aktionselementen, die gängigen Anwendungsfällen von Apache Kafka entsprechen, wie z. B. das Erzeugen und Verbrauchen von Daten, finden Sie unter the section called “Häufige Anwendungsfälle”.

Für Kafka-Versionen 2.8.0 und höher ist die WriteDataIdempotently-Berechtigung veraltet (KIP-679). `enable.idempotence = true` ist standardmäßig festgelegt. Daher bietet IAM für die Kafka-

Versionen 2.8.0 und höher nicht die gleiche Funktionalität wie Kafka-ACLs. Es ist nicht möglich, `WriteDataIdempotently` in einem Thema auszuführen, wenn nur `WriteData`-Zugriff auf dieses Thema gewährt wird. Dies hat keinen Einfluss auf den Fall, wenn `WriteData` für ALLE Themen bereitgestellt wird. In diesem Fall ist `WriteDataIdempotently` erlaubt. Dies ist auf Unterschiede in der Implementierung der IAM-Logik im Vergleich zur Implementierung der Kafka-ACLs zurückzuführen.

Um dieses Problem zu umgehen, empfehlen wir, eine Richtlinie zu verwenden, die dem folgenden Beispiel ähnelt:

```
{
    "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-1:0123456789012: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-1:0123456789012:topic/MyTestCluster/
abcd1234-0123-abcd-5678-1234abcd-1/TestTopic"
            ]
        }
    ]
}
```

In diesem Fall erlaubt `WriteData` Schreibvorgänge in `TestTopic`, während `WriteDataIdempotently` idempotente Schreibvorgänge in den Cluster erlaubt. Es ist wichtig zu beachten, dass `WriteDataIdempotently` eine Berechtigung auf Cluster-Ebene ist. Sie kann nicht auf Themenebene verwendet werden. Wenn `WriteDataIdempotently` auf die Themenebene beschränkt ist, funktioniert diese Richtlinie nicht.

Bootstrap-Broker für IAM-Zugriffssteuerung abrufen

Siehe the section called “Abrufen der Bootstrap-Broker”.

Semantik von Aktionen und Ressourcen

In diesem Abschnitt wird die Semantik der Aktions- und Ressourcenelemente erläutert, die Sie in einer IAM-Autorisierungsrichtlinie verwenden können. Eine Beispielrichtlinie finden Sie unter the section called “Autorisierungsrichtlinien erstellen”.

Aktionen

In der folgenden Tabelle sind die Aktionen aufgeführt, die Sie in eine Autorisierungsrichtlinie aufnehmen können, wenn Sie IAM-Zugriffssteuerung für Amazon MSK verwenden. Wenn Sie in Ihre Autorisierungsrichtlinie eine Aktion aus der Spalte Aktion der Tabelle aufnehmen, müssen Sie auch die entsprechenden Aktionen aus der Spalte Erforderliche Aktionen angeben.

Aktion	Beschreibung	Erforderliche Aktionen	Erforderliche - Ressourcen	Gilt für Serverless-Cluster
`kafka-cluster:Connect`	Gewährt die Berechtigung, sich mit dem Cluster zu verbinden und zu authentifizieren.	None	Cluster	Ja
`kafka-cluster:DescribeCluster`	Gewährt die Berechtigung zum Beschreiben verschiedener Aspekte des Clusters, was	`kafka-cluster:Connect`	Cluster	Ja

Aktion	Beschreibung	Erforderliche Aktionen	Erforderliche - Ressourcen	Gilt für Serverless-Cluster
	der DESCRIBE CLUSTER ACL von Apache Kafka entspricht.			

`kafka-cluster:AlterCluster`	Gewährt die Berechtigung zum Ändern verschiedener Aspekte des Clusters, was der ALTER CLUSTER ACL von Apache Kafka entspricht.	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeCluster`
			Cluster	Nein

`kafka-cluster:DescribeClusterDynamicConfiguration`	Gewährt die Berechtigung zum Beschreiben der dynamischen Konfiguration eines Clusters, was der DESCRIBE_CONFIGS CLUSTER ACL von Apache Kafka entspricht.	`kafka-cluster:Connect`	Cluster	Nein

Aktion	Beschreibung	Erforderliche Aktionen	Erforderliche - Ressourcen	Gilt für Serverless-Cluster

`kafka-cluster:AlterClusterDynamicConfiguration`	Gewährt die Berechtigung zum Ändern der dynamischen Konfiguration eines Clusters, was der ALTER_CONFIGS CLUSTER ACL von Apache Kafka entspricht.	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeClusterDynamicConfiguration`
			Cluster	Nein

`kafka-cluster:WriteDataIdempotently`	Gewährt die Berechtigung zum idempotenten Schreiben von Daten auf einen Cluster, was der IDEMPOTENT_WRITE CLUSTER ACL von Apache Kafka entspricht.	`kafka-cluster:Connect`
		
		`kafka-cluster:WriteData`
			Cluster	Ja

Aktion	Beschreibung	Erforderliche Aktionen	Erforderliche - Ressourcen	Gilt für Serverless-Cluster
`kafka-cluster:CreateTopic`	Gewährt die Berechtigung zum Erstellen von Themen auf einem Cluster, was der CREATE CLUSTER/TOPIC ACL von Apache Kafka entspricht.	`kafka-cluster:Connect`	Thema	Ja
`kafka-cluster:DescribeTopic`	Gewährt die Berechtigung zum Beschreiben von Themen auf einem Cluster, was der DESCRIBE TOPIC ACL von Apache Kafka entspricht.	`kafka-cluster:Connect`	Thema	Ja

`kafka-cluster:AlterTopic`	Gewährt die Berechtigung zum Ändern von Themen auf einem Cluster, was der ALTER TOPIC ACL von Apache Kafka entspricht.	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeTopic`
			Thema	Ja

Aktion	Beschreibung	Erforderliche Aktionen	Erforderliche - Ressourcen	Gilt für Serverless-Cluster
`kafka-cluster:DeleteTopic`	Gewährt die Berechtigung zum Löschen von Themen auf einem Cluster, was der DELETE TOPIC ACL von Apache Kafka entspricht.	`kafka-cluster:Connect` `kafka-cluster:DescribeTopic`	Thema	Ja
`kafka-cluster:DescribeTopicDynamicConfiguration`	Gewährt die Berechtigung zum Beschreiben der dynamischen Konfiguration von Themen auf einem Cluster, was der DESCRIBE_CONFIGS TOPIC ACL von Apache Kafka entspricht.	`kafka-cluster:Connect`	Thema	Ja

Aktion	Beschreibung	Erforderliche Aktionen	Erforderliche - Ressourcen	Gilt für Serverless-Cluster
`kafka-cluster:AlterTopicDynamicConfiguration`	Gewährt die Berechtigung zum Ändern der dynamischen Konfiguration von Themen auf einem Cluster, was der ALTER_CONFIGS TOPIC ACL von Apache Kafka entspricht.	`kafka-cluster:Connect` `kafka-cluster:DescribeTopicDynamicConfiguration`	Thema	Ja

`kafka-cluster:ReadData`	Gewährt die Berechtigung zum Lesen von Daten aus Themen auf einem Cluster, was der READ TOPIC ACL von Apache Kafka entspricht.	`kafka-cluster:Connect`
		`kafka-cluster:DescribeTopic`
		`kafka-cluster:AlterGroup`
			Thema	Ja

Aktion	Beschreibung	Erforderliche Aktionen	Erforderliche - Ressourcen	Gilt für Serverless-Cluster

`kafka-cluster:WriteData`	Gewährt die Berechtigung zum Schreiben von Daten zu Themen auf einem Cluster, was der WRITE-TOPIC-ACL von Apache Kafka entspricht	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeTopic`
			Thema	Ja

`kafka-cluster:DescribeGroup`	Gewährt die Berechtigung zum Beschreiben von Gruppen auf einem Cluster, was der DESCRIBE GROUP ACL von Apache Kafka entspricht.	`kafka-cluster:Connect`	Gruppe	Ja

`kafka-cluster:AlterGroup`	Gewährt die Berechtigung, Gruppen in einem Cluster beizutreten, was der READ GROUP ACL von Apache Kafka entspricht.	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeGroup`
			Gruppe	Ja

Aktion	Beschreibung	Erforderliche Aktionen	Erforderliche - Ressourcen	Gilt für Serverless-Cluster
`kafka-cluster:DeleteGroup`	Gewährt die Berechtigung zum Löschen von Gruppen auf einem Cluster, was der DELETE GROUP ACL von Apache Kafka entspricht.	`kafka-cluster:Connect` `kafka-cluster:DescribeGroup`	Gruppe	Ja
`kafka-cluster:DescribeTransactionalId`	Gewährt die Berechtigung zum Beschreiben der Transaktions-IDs auf einem Cluster, was der DESCRIBE TRANSACTIONAL_ID ACL von Apache Kafka entspricht.	`kafka-cluster:Connect`	transactional-id	Ja

Aktion	Beschreibung	Erforderliche Aktionen	Erforderliche - Ressourcen	Gilt für Serverless-Cluster

`kafka-cluster:AlterTransactionalId`	Gewährt die Berechtigung zum Ändern der Transaktions-IDs auf einem Cluster, was der WRITE TRANSACTIONAL_ID ACL von Apache Kafka entspricht.	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeTransactionalId`
		
		`kafka-cluster:WriteData`
			transactional-id	Ja

Sie können das Sternchen (*) als Platzhalter in einer Aktion hinter dem Doppelpunkt beliebig oft verwenden. Im Folgenden sind einige Beispiele aufgeführt.

- `kafka-cluster:*Topic` steht für `kafka-cluster:CreateTopic`, `kafka-cluster:DescribeTopic`, `kafka-cluster:AlterTopic` und `kafka-cluster:DeleteTopic`. Es beinhaltet nicht `kafka-cluster:DescribeTopicDynamicConfiguration` oder `kafka-cluster:AlterTopicDynamicConfiguration`.
- `kafka-cluster:*` steht für alle Berechtigungen.

Ressourcen

In der folgenden Tabelle sind die vier Arten von Ressourcen aufgeführt, die Sie in eine Autorisierungsrichtlinie aufnehmen können, wenn Sie IAM-Zugriffssteuerung für Amazon MSK verwenden. Sie können den Cluster-Amazon-Ressourcennamen (ARN) von AWS Management Console oder mithilfe der DescribeClusterAPI oder des Befehls describe-cluster AWS CLI abrufen. Anschließend können Sie den Cluster-ARN verwenden, um Themen-, Gruppen- und Transaktions-ID-ARNs zu erstellen. Um eine Ressource in einer Autorisierungsrichtlinie anzugeben, verwenden Sie den ARN dieser Ressource.

Ressource	ARN-Format
Cluster	arn:aws:kafka:*region*:*account-id* :cluster/*cluster-name* /*cluster-uuid*
Thema	arn:aws:kafka:*region*:*account-id* :topic/*cluster-name* /*cluster-uuid* /*topic-name*
Gruppe	arn:aws:kafka:*region*:*account-id* :group/*cluster-name* /*cluster-uuid* /*group-name*
Transkaktions-ID	arn:aws:kafka:*region*:*account-id* :transactional-id/*cluster-name* /*cluster-uuid* /*transactional-id*

Sie können das Sternchen (*) als Platzhalter beliebig oft an beliebiger Stelle in dem Teil des ARN verwenden, der nach `:cluster/`, `:topic/`, `:group/` und `:transactional-id/` folgt. Im Folgenden finden Sie einige Beispiele dafür, wie Sie das Sternchen (*) als Platzhalter verwenden können, um auf mehrere Ressourcen zu verweisen:

- `arn:aws:kafka:us-east-1:0123456789012:topic/MyTestCluster/*`: alle Themen in einem beliebigen Cluster mit dem Namen MyTestCluster, unabhängig von der UUID des Clusters.
- `arn:aws:kafka:us-east-1:0123456789012:topic/MyTestCluster/abcd1234-0123-abcd-5678-1234abcd-1/*_test`: alle Themen, deren Name mit „_test“ endet, in dem Cluster, dessen Name MyTestCluster und dessen UUID abcd1234-0123-abcd-5678-1234abcd-1 ist.
- `arn:aws:kafka:us-east-1:0123456789012:transactional-id/MyTestCluster/*/5555abcd-1111-abcd-1234-abcd1234-1`: alle Transaktionen, deren Transaktions-ID 5555abcd-1111-abcd-1234-abcd1234-1 lautet, in allen Inkarnationen eines Clusters, der in Ihrem Konto benannt ist. MyTestCluster Das heißt, wenn Sie einen Cluster mit dem Namen erstellen MyTestCluster, ihn dann löschen und dann einen weiteren Cluster mit demselben Namen erstellen, können Sie diesen Ressourcen-ARN verwenden, um dieselbe Transaktions-ID auf beiden Clustern darzustellen. Auf den gelöschten Cluster kann jedoch nicht zugegriffen werden.

Häufige Anwendungsfälle

Die erste Spalte der folgenden Tabelle zeigt einige gängige Anwendungsfälle. Um einen Client zur Ausführung eines bestimmten Anwendungsfalls zu autorisieren, nehmen Sie die für diesen

Anwendungsfall erforderlichen Aktionen in die Autorisierungsrichtlinie des Clients auf und stellen Sie `Effect` auf `Allow` ein.

Informationen zu allen Aktionen, die Teil der IAM-Zugriffssteuerung für Amazon MSK sind, finden Sie unter the section called “Semantik von Aktionen und Ressourcen”.

> **Note**
>
> Aktionen werden standardmäßig verweigert. Sie müssen jede Aktion, zu deren Ausführung Sie den Client autorisieren möchten, ausdrücklich erlauben.

Anwendungsfall	Erforderliche Aktionen
Admin.	`kafka-cluster:*`
Erstellen eines Themas	`kafka-cluster:Connect` `kafka-cluster:CreateTopic`

Daten produzieren	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:WriteData`

Daten verbrauchen	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:DescribeGroup`
	`kafka-cluster:AlterGroup`
	`kafka-cluster:ReadData`

Daten idempotent produzieren	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:WriteData`

Anwendungsfall	Erforderliche Aktionen
	`kafka-cluster:WriteDataIdempotently`

Daten transaktionell produzieren	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:WriteData`
	`kafka-cluster:DescribeTransactionalId`
	`kafka-cluster:AlterTransactionalId`

Die Konfiguration eines Clusters beschreiben	`kafka-cluster:Connect` `kafka-cluster:DescribeClusterDynamicConfiguration`

Die Konfiguration eines Clusters aktualisieren	`kafka-cluster:Connect`
	`kafka-cluster:DescribeClusterDynamicConfiguration`
	`kafka-cluster:AlterClusterDynamicConfiguration`

Die Konfiguration eines Themas beschreiben	`kafka-cluster:Connect` `kafka-cluster:DescribeTopicDynamicConfiguration`

Anwendungsfall	Erforderliche Aktionen

Die Konfiguration eines Themas aktualisieren	`kafka-cluster:Connect`
	
	`kafka-cluster:DescribeTopic DynamicConfiguration`
	
	`kafka-cluster:AlterTopicDyn amicConfiguration`

Ein Thema ändern	`kafka-cluster:Connect`
	
	`kafka-cluster:DescribeTopic`
	
	`kafka-cluster:AlterTopic`

Gegenseitige TLS-Authentifizierung

Sie können die Client-Authentifizierung mit TLS für Verbindungen von Ihren Anwendungen zu Ihren Amazon MSK-Brokern aktivieren. Damit Sie die Client-Authentifizierung verwenden können, benötigen Sie eine AWS Private CA. AWS Private CA Sie können sich entweder in demselben AWS-Konto Cluster oder in einem anderen Konto befinden. Informationen zu AWS Private CA s finden Sie unter Erstellen und Verwalten von AWS Private CA.

> **Note**
>
> TLS-Authentifizierung ist derzeit in den Regionen Peking und Ningxia nicht verfügbar.

Amazon MSK unterstützt keine Zertifikatswiderrufslisten (CRLs). Verwenden Sie Apache Kafka ACLs und Sicherheitsgruppen, um den Zugriff auf Ihre Cluster-Themen zu kontrollieren oder kompromittierte Zertifikate zu blockieren. AWS Informationen zur Verwendung von Apache-Kafka-ACLs finden Sie unter the section called “Apache Kafka ACLs”.

Dieses Thema enthält die folgenden Abschnitte:

- Erstellen eines Cluster, der die Client-Authentifizierung unterstützt
- Einrichten eines Clients zur Verwendung der Authentifizierung
- Erstellen und Verwenden von Nachrichten mithilfe der Authentifizierung

Erstellen eines Cluster, der die Client-Authentifizierung unterstützt

Dieses Verfahren zeigt Ihnen, wie Sie die Client-Authentifizierung mithilfe von aktivieren. AWS Private CA

> **Note**
>
> Wir empfehlen dringend, unabhängig AWS Private CA für jeden MSK-Cluster zu verwenden, wenn Sie Mutual TLS zur Zugriffskontrolle verwenden. Dadurch wird sichergestellt, dass von PCAs signierte TLS-Zertifikate nur bei einem einzigen MSK-Cluster authentifiziert werden.

1. Erstellen Sie eine Datei mit dem Namen `clientauthinfo.json` und dem folgenden Inhalt. Ersetzen Sie *`Private-CA-ARN`* durch den ARN Ihrer PCA.

   ```
   {
      "Tls": {
          "CertificateAuthorityArnList": ["Private-CA-ARN"]
       }
   }
   ```

2. Erstellen Sie eine Datei mit dem Namen `brokernodegroupinfo.json`, wie unter the section called “Erstellen eines Clusters mit dem AWS CLI” beschrieben.
3. Für die Client-Authentifizierung müssen Sie auch die Verschlüsselung während der Übertragung zwischen Clients und Brokern aktivieren. Erstellen Sie eine Datei mit dem Namen `encryptioninfo.json` und dem folgenden Inhalt. Ersetzen Sie *`KMS-Key-ARN`* durch den ARN Ihres KMS-Schlüssels. Für `ClientBroker` können Sie `TLS` oder `TLS_PLAINTEXT` festlegen.

   ```
   {
      "EncryptionAtRest": {
          "DataVolumeKMSKeyId": "KMS-Key-ARN"
       },
      "EncryptionInTransit": {
           "InCluster": true,
           "ClientBroker": "TLS"
       }
   }
   ```

   Weitere Informationen zur Verschlüsselung finden Sie unter the section called “Verschlüsselung”.

4. Führen Sie auf einem Computer, auf dem Sie das AWS CLI installiert haben, den folgenden Befehl aus, um einen Cluster mit aktivierter Authentifizierung und Verschlüsselung bei der Übertragung zu erstellen. Speichern Sie den in der Antwort angegebenen Cluster-ARN.

```
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
```

Einrichten eines Clients zur Verwendung der Authentifizierung

1. Erstellen Sie eine Amazon-EC2-Instance, die als Client-Computer verwendet werden soll. Erstellen Sie diese Instance der Einfachheit halber in derselben VPC, die Sie für den Cluster verwendet haben. Unter the section called “Schritt 3: Einen Client-Computer erstellen” finden Sie ein Beispiel dafür, wie Sie solch einen Client-Computer erstellen können.
2. Erstellen eines Themas. Ein Beispiel finden Sie in den Anweisungen unter the section called “Schritt 4: Ein Thema erstellen”.
3. Führen Sie auf einem Computer, auf dem Sie das AWS CLI installiert haben, den folgenden Befehl aus, um die Bootstrap-Broker des Clusters abzurufen. Ersetzen Sie *Cluster-ARN* durch den ARN Ihres Clusters.

```
aws kafka get-bootstrap-brokers --cluster-arn Cluster-ARN
```

   Speichern Sie die Zeichenfolge, die `BootstrapBrokerStringTls` in der Antwort zugeordnet ist.
4. Führen Sie auf Ihrem Client-Computer den folgenden Befehl aus, um mithilfe des JVM-Vertrauensspeichers Ihren Client-Vertrauensspeicher zu erstellen. Wenn Ihr JVM-Pfad anders ist, passen Sie den Befehl entsprechend an.

```
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. Führen Sie auf Ihrem Client-Computer den folgenden Befehl aus, um einen privaten Schlüssel für Ihren Client zu erstellen. Ersetzen Sie *Distinguished-Name*, *Example-Alias*, *Your-Store-Pass* und *Your-Key-Pass* durch Zeichenfolgen Ihrer Wahl.

```
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
```

6. Führen Sie auf Ihrem Client-Computer den folgenden Befehl aus, um eine Zertifikatsanforderung mit dem privaten Schlüssel zu erstellen, den Sie im vorherigen Schritt erstellt haben.

```
keytool -keystore kafka.client.keystore.jks -certreq -file client-cert-sign-request
 -alias Example-Alias -storepass Your-Store-Pass -keypass Your-Key-Pass
```

7. Öffnen Sie die Datei `client-cert-sign-request`, und stellen Sie sicher, dass sie mit `-----BEGIN CERTIFICATE REQUEST-----` beginnt und mit `-----END CERTIFICATE REQUEST-----` endet. Wenn sie mit `-----BEGIN NEW CERTIFICATE REQUEST-----` beginnt , löschen Sie das Wort `NEW` (und das einzelne Leerzeichen, das darauf folgt) vom Anfang und vom Ende der Datei.

8. Führen Sie auf einem Computer, auf dem Sie das AWS CLI installiert haben, den folgenden Befehl aus, um Ihre Zertifikatsanforderung zu signieren. Ersetzen Sie *Private-CA-ARN* durch den ARN Ihrer PCA. Sie können den Gültigkeitswert ändern, wenn Sie möchten. Hier verwenden wir 300 als Beispiel.

```
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"
```

   Speichern Sie den in der Antwort angegebenen Zertifikat-ARN.

   > **Note**
   >
   > Um Ihr Client-Zertifikat abzurufen, verwenden Sie den Befehl `acm-pca get-certificate` und geben Sie Ihren Zertifikat-ARN an. Weitere Informationen finden Sie unter get-certificate in der AWS CLI -Befehlsreferenz.

9. Führen Sie den folgenden Befehl aus, um das Zertifikat abzurufen, das für Sie AWS Private CA signiert wurde. Ersetzen Sie *Certificate-ARN* durch den ARN, den Sie in der Antwort auf den vorherigen Befehl erhalten haben.

```
aws acm-pca get-certificate --certificate-authority-arn Private-CA-ARN --
certificate-arn Certificate-ARN
```

10. Kopieren Sie aus dem JSON-Ergebnis der Ausführung des vorherigen Befehls die Zeichenfolgen, die `Certificate` und `CertificateChain` zugeordnet sind. Fügen Sie diese beiden Zeichenfolgen in eine neue Datei mit dem Namen ein signed-certificate-from-acm. Fügen Sie die Zeichenfolge, die `Certificate` zugeordnet ist, zuerst ein, gefolgt von der Zeichenfolge, die `CertificateChain` zugeordnet ist. Ersetzen Sie die Zeichen `\n` durch neue Zeilen. Im Folgenden finden Sie die Struktur der Datei, nachdem Sie das Zertifikat und die Zertifikatkette eingefügt haben.

```
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
```

11. Führen Sie den folgenden Befehl auf dem Client-Computer aus, um dieses Zertifikat zu Ihrem Schlüsselspeicher hinzuzufügen, damit Sie es bei der Kommunikation mit den MSK-Brokern bereitstellen können.

```
keytool -keystore kafka.client.keystore.jks -import -file signed-certificate-from-
acm -alias Example-Alias -storepass Your-Store-Pass -keypass Your-Key-Pass
```

12. Erstellen Sie eine Datei mit dem Namen `client.properties` und dem folgenden Inhalt. Passen Sie die Speicherorte des Vertrauensspeichers und des Schlüsselspeichers an die Pfade an, in denen Sie `kafka.client.truststore.jks` gespeichert haben. Ersetzen Sie den Platzhalter *{YOUR KAFKA VERSION}* durch Ihre Kafka-Client-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
```

Erstellen und Verwenden von Nachrichten mithilfe der Authentifizierung

1. Führen Sie den folgenden Befehl aus, um ein Thema zu erstellen. Die Datei namens `client.properties` ist die Datei, die Sie im vorherigen Verfahren erstellt haben.

   ```
   <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. Führen Sie den folgenden Befehl aus, um einen Konsolenproduzenten zu starten. Die Datei namens `client.properties` ist die Datei, die Sie im vorherigen Verfahren erstellt haben.

   ```
   <path-to-your-kafka-installation>/bin/kafka-console-producer.sh --bootstrap-
   server BootstrapBroker-String --topic ExampleTopic --producer.config
    client.properties
   ```

3. Führen Sie auf Ihrem Client-Computer in einem neuen Befehlsfenster den folgenden Befehl aus, um einen Konsolenverbraucher zu starten.

   ```
   <path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
   server BootstrapBroker-String --topic ExampleTopic --consumer.config
    client.properties
   ```

4. Geben Sie Nachrichten in das Produzentenfenster ein und beobachten Sie, wie sie im Verbraucherfenster angezeigt werden.

Authentifizierung der Anmeldedaten mit AWS Secrets Manager

Sie können den Zugriff auf Ihre Amazon MSK-Cluster mithilfe von Anmeldeinformationen steuern, die mit AWS Secrets Manager gespeichert und gesichert werden. Das Speichern von Benutzeranmeldeinformationen in Secrets Manager reduziert den Aufwand für die Cluster-Authentifizierung, wie z. B. die Prüfung, Aktualisierung und Rotation von Anmeldeinformationen. Mit Secrets Manager können Sie auch Benutzeranmeldeinformationen clusterübergreifend freigeben.

Dieses Thema enthält die folgenden Abschnitte:

- Funktionsweise

- Einrichtung der SASL/SCRAM-Authentifizierung für einen Amazon-MSK-Cluster
- Working with users
- Einschränkungen

Funktionsweise

Die Authentifizierung über Anmeldeinformationen für Amazon MSK verwendet SASL/SCRAM (Simple Authentication and Security Layer/Salted Challenge Response Mechanism)-Authentifizierung. Um die Authentifizierung über Anmeldeinformationen für einen Cluster einzurichten, erstellen Sie eine Secret-Ressource in AWS Secrets Manager und ordnen diesem Secret Anmeldeinformationen zu.

SASL/SCRAM ist in RFC 5802 definiert. SCRAM verwendet gesicherte Hashing-Algorithmen und überträgt keine Klartext-Anmeldeinformationen zwischen dem Client und dem Server.

> **Note**
>
> Wenn Sie die SASL/SCRAM-Authentifizierung für Ihren Cluster einrichten, aktiviert Amazon MSK die TLS-Verschlüsselung für den gesamten Datenverkehr zwischen Clients und Brokern.

Einrichtung der SASL/SCRAM-Authentifizierung für einen Amazon-MSK-Cluster

Um ein Geheimnis in AWS Secrets Manager einzurichten, folgen Sie dem Tutorial Creating and Retrieving a Secret im AWS Secrets Manager Manager-Benutzerhandbuch.

Beachten Sie die folgenden Anforderungen, wenn Sie ein Secret für einen Amazon-MSK-Cluster erstellen:

- Wählen Sie für Secret-Typ die Option Anderer Secret-Typ (z. B. API-Schlüssel).
- Ihr Secret-Nname muss mit dem Präfix AmazonMSK_ beginnen.
- Sie müssen entweder einen vorhandenen benutzerdefinierten AWS KMS Schlüssel verwenden oder einen neuen benutzerdefinierten AWS KMS Schlüssel für Ihr Geheimnis erstellen. Secrets Manager verwendet standardmäßig den AWS KMS Standardschlüssel für ein Geheimnis.

> **Important**
>
> Ein mit dem AWS KMS Standardschlüssel erstelltes Geheimnis kann nicht mit einem Amazon MSK-Cluster verwendet werden.

- Ihre Anmeldeinformationen müssen das folgende Format haben, um Schlüssel-Wert-Paare mit der Klartext-Option eingeben zu können.

```
{
  "username": "alice",
  "password": "alice-secret"
}
```

- Notieren Sie sich den ARN (Amazon-Ressourcenname) für Ihr Secret.
-

> **Important**
>
> Sie können einem Cluster, der die unter the section called “ Die Größe Ihres Clusters anpassen: Anzahl der Partitionen pro Broker” beschriebenen Grenzwerte überschreitet, kein Secrets-Manager-Secret zuordnen.

- Wenn Sie den AWS CLI zum Erstellen des Geheimnisses verwenden, geben Sie eine Schlüssel-ID oder einen ARN für den `kms-key-id` Parameter an. Geben Sie keinen Alias an.
- Um das Geheimnis Ihrem Cluster zuzuordnen, verwenden Sie entweder die Amazon MSK-Konsole oder den BatchAssociateScramSecretVorgang.

> **Important**
>
> Wenn Sie einem Cluster ein Secret zuordnen, fügt Amazon MSK dem Secret eine Ressourcenrichtlinie hinzu, die es Ihrem Cluster ermöglicht, auf die von Ihnen definierten geheimen Werte zuzugreifen und diese zu lesen. Sie sollten diese Ressourcenrichtlinie nicht ändern. Andernfalls kann Ihr Cluster daran gehindert werden, auf Ihr Secret zuzugreifen.

Die folgende Beispiel-JSON-Eingabe für den Vorgang `BatchAssociateScramSecret` ordnet ein Secret einem Cluster zu:

```
{
  "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"
  ]
}
```

Herstellen einer Verbindung zu Ihrem Cluster mit Anmeldeinformationen

Nachdem Sie ein Secret erstellt und es Ihrem Cluster zugeordnet haben, können Sie Ihren Client mit dem Cluster verbinden. Die folgenden Beispielschritte zeigen, wie Sie einen Client mit einem Cluster verbinden, der die SASL/SCRAM-Authentifizierung verwendet, und wie Sie aus einem Beispielthema produzieren und verbrauchen.

1. Führen Sie den folgenden Befehl auf einem Computer aus, auf dem die AWS CLI installiert ist, und ersetzen Sie *`ClusterArn`* durch den ARN Ihres Clusters.

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

2. Um ein Beispielthema zu erstellen, führen Sie den folgenden Befehl aus und ersetzen Sie *`BootstrapServerString`* durch einen der Broker-Endpunkte, die Sie im vorherigen Schritt abgerufen haben.

   ```
   <path-to-your-kafka-installation>/bin/kafka-topics.sh --create --bootstrap-
   server BootstrapServerString --replication-factor 3 --partitions 1 --topic
    ExampleTopicName
   ```

3. Erstellen Sie auf Ihrem Client-Computer eine JAAS-Konfigurationsdatei, die die in Ihrem Secret gespeicherten Benutzeranmeldeinformationen enthält. Erstellen Sie beispielsweise für den Benutzer alice eine Datei namens `users_jaas.conf` mit dem folgenden Inhalt.

   ```
   KafkaClient {
      org.apache.kafka.common.security.scram.ScramLoginModule required
      username="alice"
      password="alice-secret";
   };
   ```

4. Verwenden Sie den folgenden Befehl, um Ihre JAAS-Konfigurationsdatei als KAFKA_OPTS-Umgebungsparameter zu exportieren.

```
export KAFKA_OPTS=-Djava.security.auth.login.config=<path-to-jaas-file>/
users_jaas.conf
```

5. Erstellen Sie in einem ./tmp-Verzeichnis eine Datei namens kafka.client.truststore.jks.

6. Verwenden Sie den folgenden Befehl, um die JDK-Schlüsselspeicherdatei aus Ihrem JVM-cacerts-Ordner in die kafka.client.truststore.jks-Datei zu kopieren, die Sie im vorherigen Schritt erstellt haben. Ersetzen Sie *JDKFolder* durch den Namen des JDK-Ordners auf Ihrer Instance. Beispielsweise könnte Ihr JDK-Ordner java-1.8.0-openjdk-1.8.0.201.b09-0.amzn2.x86_64 benannt sein.

```
cp /usr/lib/jvm/JDKFolder/jre/lib/security/cacerts /tmp/kafka.client.truststore.jks
```

7. Erstellen Sie im bin-Verzeichnis Ihrer Apache-Kafka-Installation eine Client-Eigenschaftendatei namens client_sasl.properties mit dem folgenden Inhalt. Diese Datei definiert den SASL-Mechanismus und das SASL-Protokoll.

```
security.protocol=SASL_SSL
sasl.mechanism=SCRAM-SHA-512
ssl.truststore.location=<path-to-keystore-file>/kafka.client.truststore.jks
```

8. Rufen Sie die Zeichenfolge Ihres Bootstrap-Brokers mit dem folgenden Befehl ab. *ClusterArn*Ersetzen Sie durch den Amazon-Ressourcennamen (ARN) Ihres Clusters:

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

   Speichern Sie aus dem JSON-Ergebnis des Befehls den Wert, der der Zeichenfolge BootstrapBrokerStringSaslScram zugeordnet ist.

9. Führen Sie den folgenden Befehl auf Ihrem Client-Computer aus, um in dem von Ihnen erstellten Beispielthema zu produzieren. Ersetzen Sie *BootstrapBrokerStringSaslScram* durch den Wert, den Sie im vorherigen Schritt abgerufen haben.

```
<path-to-your-kafka-installation>/bin/kafka-console-producer.sh --broker-
list BootstrapBrokerStringSaslScram --topic ExampleTopicName --producer.config
 client_sasl.properties
```

10. Führen Sie den folgenden Befehl auf Ihrem Client-Computer aus, um aus dem von Ihnen erstellten Thema zu verbrauchen. Ersetzen Sie *BootstrapBrokerStringSaslScram* durch den Wert, den Sie zuvor erhalten haben.

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBrokerStringSaslScram --topic ExampleTopicName --from-beginning --
consumer.config client_sasl.properties
```

Working with users

Benutzer erstellen: Sie erstellen Benutzer in Ihrem Secret als Schlüssel-Wert-Paare. Wenn Sie die Klartext-Option in der Secrets-Manager-Konsole verwenden, sollten Sie die Anmeldeinformationen im folgenden Format angeben.

```
{
  "username": "alice",
  "password": "alice-secret"
}
```

Benutzerzugriff widerrufen: Um die Anmeldeinformationen eines Benutzers für den Zugriff auf einen Cluster zu widerrufen, empfehlen wir, zuerst eine ACL für den Cluster zu entfernen oder zu erzwingen und dann die Zuordnung des Secrets aufzuheben. Dies ist auf Folgendes zurückzuführen:

- Durch das Entfernen eines Benutzers werden bestehende Verbindungen nicht geschlossen.
- Es dauert bis zu 10 Minuten, bis Änderungen an Ihrem Secret verbreitet sind.

Weitere Informationen zur Verwendung einer ACL mit Amazon MSK finden Sie unter Apache Kafka ACLs.

Für Cluster, die ZooKeeper den Modus verwenden, empfehlen wir, den Zugriff auf Ihre ZooKeeper Knoten einzuschränken, um zu verhindern, dass Benutzer ACLs ändern. Weitere Informationen finden Sie unter Steuern des Zugriffs auf Apache ZooKeeper.

Einschränkungen

Beachten Sie die folgenden Einschränkungen bei der Verwendung von SCRAM-Secrets:

- Amazon MSK unterstützt nur SCRAM-SHA-512-Authentifizierung.

- Ein Amazon-MSK-Cluster kann bis zu 1 000 Benutzer haben.
- Sie müssen eine AWS KMS key mit Ihrem Secret verwenden. Sie können kein Secret verwenden, das den standardmäßigen Secrets-Manager-Verschlüsselungsschlüssel mit Amazon MSK verwendet. Weitere Informationen zum Erstellen eines KMS-Schlüssels finden Sie unter Erstellen von symmetrischen KMS-Verschlüsselungsschlüsseln.
- Sie können keinen asymmetrischen KMS-Schlüssel mit Secrets Manager verwenden.
- Mithilfe dieser BatchAssociateScramSecretOperation können Sie einem Cluster bis zu 10 Geheimnisse gleichzeitig zuordnen.
- Der Name von Secrets, die einem Amazon-MSK-Cluster zugeordnet sind, muss das Präfix AmazonMSK_ haben.
- Mit einem Amazon MSK-Cluster verknüpfte Geheimnisse müssen sich im selben Amazon Web Services Services-Konto und derselben AWS Region wie der Cluster befinden.

Apache Kafka ACLs

Apache Kafka verfügt über einen austauschbaren Authorizer und wird mit einer Authorizer-Implementierung ausgeliefert. out-of-box Amazon MSK aktiviert diesen Autorisierer in der Datei `server.properties` auf den Brokern.

Apache Kafka-ACLs haben das Format „Principal P ist [Allowed/Denied] Operation O From Host H on any Resource R matching RP“. ResourcePattern Wenn RP nicht mit einer bestimmten Ressource „R“ übereinstimmt, hat „R“ keine zugeordneten ACLs, weshalb ausschließlich Superuser auf „R“ zugreifen dürfen. Um dieses Verhalten von Apache Kafka zu ändern, legen Sie die Eigenschaft `allow.everyone.if.no.acl.found` auf „true“ fest. Amazon MSK setzt es standardmäßig auf true. Dies bedeutet, dass bei Amazon MSK-Clustern, sofern Sie nicht explizit ACLs für eine Ressource festlegen, alle Prinzipale auf diese Ressource zugreifen können. Wenn Sie ACLs für eine Ressource aktivieren, können nur die autorisierten Prinzipale darauf zugreifen. Wenn Sie den Zugriff auf ein Thema einschränken und einen Client mithilfe der gegenseitigen TLS-Authentifizierung autorisieren möchten, fügen Sie ACLs mit der Apache Kafka-Autorisierungs-CLI hinzu. Weitere Informationen zum Hinzufügen, Entfernen und Auflisten von ACLs finden Sie unter Kafka Authorization Command Line Interface.

Zusätzlich zum Client müssen Sie allen Brokern Zugriff auf Ihre Themen gewähren, damit die Broker Nachrichten von der primären Partition replizieren können. Wenn die Broker keinen Zugriff auf ein Thema haben, schlägt die Replikation für das Thema fehl.

Hinzufügen oder Entfernen von Lese- und Schreibzugriff für ein Thema

1. Fügen Sie die Broker der ACL-Tabelle hinzu, damit sie aus allen Themen lesen können, in denen ACLs vorhanden sind. Um Ihren Brokern Lesezugriff auf ein Thema zu gewähren, führen Sie den folgenden Befehl auf einem Client-Computer aus, der mit dem MSK-Cluster kommunizieren kann.

   Ersetzen Sie *Distinguished-Name* durch den DNS eines Bootstrap-Brokers Ihres Clusters und ersetzen Sie dann die Zeichenfolge vor dem ersten Punkt in diesem Distinguished Name durch ein Sternchen (*). Wenn z. B. einer der Bootstrap-Broker Ihres Clusters über den DNS `b-6.mytestcluster.67281x.c4.kafka.us-east-1.amazonaws.com` verfügt, ersetzen Sie *Distinguished-Name* im folgenden Befehl durch `*.mytestcluster.67281x.c4.kafka.us-east-1.amazonaws.com`. Informationen zum Abrufen der Bootstrap-Broker finden Sie unter the section called “Abrufen der Bootstrap-Broker”.

   ```
   <path-to-your-kafka-installation>/bin/kafka-acls.sh --authorizer-properties
    --bootstrap-server BootstrapServerString --add --allow-principal
    "User:CN=Distinguished-Name" --operation Read --group=* --topic Topic-Name
   ```

2. Zum Gewähren von Lesezugriff auf ein Thema führen Sie den folgenden Befehl auf Ihrem Client-Computer aus. Wenn Sie gegenseitige TLS-Authentifizierung benutzen, verwenden Sie denselben *Distinguished-Name*, den Sie beim Erstellen des privaten Schlüssels verwendet haben.

   ```
   <path-to-your-kafka-installation>/bin/kafka-acls.sh --authorizer-properties
    --bootstrap-server BootstrapServerString --add --allow-principal
    "User:CN=Distinguished-Name" --operation Read --group=* --topic Topic-Name
   ```

   Zum Entfernen des Lesezugriffs können Sie denselben Befehl ausführen und `--add` durch `--remove` ersetzen.

3. Zum Gewähren von Schreibzugriff auf ein Thema führen Sie den folgenden Befehl auf Ihrem Client-Computer aus. Wenn Sie gegenseitige TLS-Authentifizierung benutzen, verwenden Sie denselben *Distinguished-Name*, den Sie beim Erstellen des privaten Schlüssels verwendet haben.

```
<path-to-your-kafka-installation>/bin/kafka-acls.sh --authorizer-properties
 --bootstrap-server BootstrapServerString --add --allow-principal
 "User:CN=Distinguished-Name" --operation Write --topic Topic-Name
```

Zum Entfernen des Schreibzugriffs können Sie denselben Befehl ausführen und `--add` durch `--remove` ersetzen.

Ändern der Sicherheitsgruppe eines Amazon-MSK-Clusters

Auf dieser Seite wird erklärt, wie Sie die Sicherheitsgruppe eines vorhandenen MSK-Clusters ändern. Möglicherweise müssen Sie die Sicherheitsgruppe eines Clusters ändern, um einer bestimmten Gruppe von Benutzern Zugriff zu gewähren oder den Zugriff auf den Cluster einzuschränken. Weitere Informationen zu Sicherheitsgruppen finden Sie unter Sicherheitsgruppen für Ihre VPC im Amazon-VPC-Benutzerhandbuch.

1. Verwenden Sie die ListNodesAPI oder den Befehl list-nodes in der, AWS CLI um eine Liste der Broker in Ihrem Cluster abzurufen. Die Ergebnisse dieses Vorgangs beinhalten die IDs der Elastic-Network-Schnittstellen (ENIs), die den Brokern zugeordnet sind.
2. Melden Sie sich bei der Amazon EC2 EC2-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/ec2/.
3. Wählen Sie im Dropdown-Menü in der oberen rechten Ecke des Bildschirms die Region aus, in der der Cluster bereitgestellt wird.
4. Wählen Sie im linken Bereich unter Netzwerk und Sicherheit die Option Netzwerkschnittstellen.
5. Wählen Sie die erste ENI aus, die Sie im ersten Schritt erhalten haben. Wählen Sie oben auf dem Bildschirm das Menü Aktionen und anschließend Sicherheitsgruppen ändern. Weisen Sie dieser ENI die neue Sicherheitsgruppe zu. Wiederholen Sie diesen Schritt für jede der ENIs, die Sie im ersten Schritt erhalten haben.

> **Note**
>
> Änderungen, die Sie mit der Amazon-EC2-Konsole an der Sicherheitsgruppe eines Clusters vornehmen, werden nicht in der MSK-Konsole unter Netzwerkeinstellungen wiedergegeben.

6. Konfigurieren Sie die Regeln der neuen Sicherheitsgruppe, um sicherzustellen, dass Ihre Clients Zugriff auf die Broker haben. Weitere Informationen zum Einrichten von Regeln für Sicherheitsgruppen finden Sie unter Hinzufügen, Entfernen und Aktualisieren von Regeln im Amazon-VPC-Benutzerhandbuch.

> **Important**
>
> Wenn Sie die Sicherheitsgruppe ändern, die den Brokern eines Clusters zugeordnet ist, und diesem Cluster dann neue Broker hinzufügen, ordnet Amazon MSK die neuen Broker der ursprünglichen Sicherheitsgruppe zu, die dem Cluster zugeordnet war, als der Cluster erstellt wurde. Damit ein Cluster jedoch ordnungsgemäß funktioniert, müssen alle seine Broker derselben Sicherheitsgruppe zugeordnet sein. Wenn Sie also nach dem Ändern der Sicherheitsgruppe neue Broker hinzufügen, müssen Sie das vorherige Verfahren erneut ausführen und die ENIs der neuen Broker aktualisieren.

Steuern des Zugriffs auf Apache ZooKeeper

Aus Sicherheitsgründen können Sie den Zugriff auf die ZooKeeper Apache-Knoten einschränken, die Teil Ihres Amazon MSK-Clusters sind. Zum Beschränken des Zugriffs auf die Knoten können Sie ihnen eine separate Sicherheitsgruppe zuweisen. Anschließend können Sie entscheiden, wer Zugriff auf diese Sicherheitsgruppe erhält.

> **Important**
>
> Dieser Abschnitt gilt nicht für Cluster, die im KraFT-Modus ausgeführt werden. Siehe the section called “KraFt-Modus ”.

Dieses Thema enthält die folgenden Abschnitte:

- Um Ihre ZooKeeper Apache-Knoten in einer separaten Sicherheitsgruppe zu platzieren
- Verwendung der TLS-Sicherheit mit Apache ZooKeeper

Um Ihre ZooKeeper Apache-Knoten in einer separaten Sicherheitsgruppe zu platzieren

1. Rufen Sie die ZooKeeper Apache-Verbindungszeichenfolge für Ihren Cluster ab. Um zu erfahren wie dies geht, vgl. the section called “ZooKeeper Modus”. Die Verbindungszeichenfolge enthält die DNS-Namen Ihrer ZooKeeper Apache-Knoten.
2. Verwenden Sie ein Tool wie `host` oder `ping`, um die DNS-Namen, die Sie im vorherigen Schritt erhalten haben, in IP-Adressen zu konvertieren. Speichern Sie diese IP-Adressen, da Sie sie später in diesem Verfahren benötigen.
3. Melden Sie sich bei der Amazon EC2 EC2-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/ec2/.
4. Klicken Sie im linken Bereich unter NETWORK & SECURITY (NETZWERK UND SICHERHEIT) auf Network Interfaces (Netzwerkschnittstellen).
5. Geben Sie im Suchfeld über der Tabelle der Netzwerkschnittstellen den Namen des Clusters ein, und geben Sie dann „return“ ein. Dadurch wird die Anzahl der Netzwerkschnittstellen, die in der Tabelle angezeigt werden, auf die Schnittstellen beschränkt, die dem Cluster zugeordnet sind.
6. Aktivieren Sie das Kontrollkästchen am Anfang der Zeile, die der ersten Netzwerkschnittstelle in der Liste entspricht.
7. Suchen Sie im Detailbereich unten auf der Seite nach der primären privaten IPv4-IP. Wenn diese IP-Adresse mit einer der IP-Adressen übereinstimmt, die Sie im ersten Schritt dieses Verfahrens erhalten haben, bedeutet dies, dass diese Netzwerkschnittstelle einem ZooKeeper Apache-Knoten zugewiesen ist, der Teil Ihres Clusters ist. Andernfalls deaktivieren Sie das Kontrollkästchen neben dieser Netzwerkschnittstelle, und wählen Sie die nächste Netzwerkschnittstelle in der Liste aus. Die Reihenfolge, in der Sie die Netzwerkschnittstellen auswählen, spielt keine Rolle. In den nächsten Schritten führen Sie nacheinander dieselben Operationen an allen Netzwerkschnittstellen durch, die ZooKeeper Apache-Knoten zugewiesen sind.
8. Wenn Sie eine Netzwerkschnittstelle auswählen, die einem ZooKeeper Apache-Knoten entspricht, wählen Sie oben auf der Seite das Menü Aktionen und dann Sicherheitsgruppen ändern. Weisen Sie dieser Netzwerkschnittstelle eine neue Sicherheitsgruppe zu. Weitere Informationen zum Erstellen von Sicherheitsgruppen finden Sie unter Erstellen einer Sicherheitsgruppe in der Amazon-VPC-Dokumentation.
9. Wiederholen Sie den vorherigen Schritt, um allen Netzwerkschnittstellen, die den ZooKeeper Apache-Knoten Ihres Clusters zugeordnet sind, dieselbe neue Sicherheitsgruppe zuzuweisen.

10. Nun können Sie auswählen, wer Zugriff auf diese neue Sicherheitsgruppe hat. Weitere Informationen zum Einrichten von Regeln für Sicherheitsgruppen finden Sie unter Hinzufügen, Entfernen und Aktualisieren von Regeln in der Amazon-VPC-Dokumentation.

Verwendung der TLS-Sicherheit mit Apache ZooKeeper

Sie können die TLS-Sicherheit für die Verschlüsselung bei der Übertragung zwischen Ihren Clients und Ihren ZooKeeper Apache-Knoten verwenden. Gehen Sie wie folgt vor, um die TLS-Sicherheit mit Ihren ZooKeeper Apache-Knoten zu implementieren:

- Cluster müssen Apache Kafka Version 2.5.1 oder höher verwenden, um TLS-Sicherheit mit Apache verwenden zu können. ZooKeeper
- Aktivieren Sie die TLS-Sicherheit, wenn Sie Ihren Cluster erstellen oder konfigurieren. Cluster, die mit Apache Kafka Version 2.5.1 oder höher und aktiviertem TLS erstellt wurden, verwenden automatisch TLS-Sicherheit mit Apache-Endpunkten. ZooKeeper Weitere Informationen zur Einrichtung von TLS-Sicherheit finden Sie unter Wie kann ich mit der Verschlüsselung beginnen?.
- Rufen Sie die TLS-Apache ZooKeeper Endpoints mithilfe des Vorgangs ab. DescribeCluster
- Erstellen Sie eine ZooKeeper Apache-Konfigurationsdatei zur Verwendung mit den `kafka-acls.sh`Tools `kafka-configs.sh` und oder mit der ZooKeeper Shell. Bei jedem Tool verwenden Sie den `--zk-tls-config-file` Parameter, um Ihre ZooKeeper Apache-Konfiguration anzugeben.

  Das folgende Beispiel zeigt eine typische ZooKeeper Apache-Konfigurationsdatei:

  ```
  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
  ```

- Für andere Befehle (z. B.`kafka-topics`) müssen Sie die `KAFKA_OPTS` Umgebungsvariable verwenden, um ZooKeeper Apache-Parameter zu konfigurieren. Das folgende Beispiel zeigt, wie die `KAFKA_OPTS` Umgebungsvariable so konfiguriert wird, dass ZooKeeper Apache-Parameter an andere Befehle übergeben werden:

  ```
  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"
```

Nachdem Sie die KAFKA_OPTS-Umgebungsvariable konfiguriert haben, können Sie CLI-Befehle normal verwenden. Im folgenden Beispiel wird mithilfe der ZooKeeper Apache-Konfiguration aus der KAFKA_OPTS Umgebungsvariablen ein Apache Kafka-Thema erstellt:

```
<path-to-your-kafka-installation>/bin/kafka-topics.sh --create --
zookeeper ZooKeeperTLSConnectString --replication-factor 3 --partitions 1 --topic
 AWSKafkaTutorialTopic
```

> **Note**
>
> Die Namen der Parameter, die Sie in Ihrer ZooKeeper Apache-Konfigurationsdatei verwenden, und der Parameter, die Sie in Ihrer KAFKA_OPTS Umgebungsvariablen verwenden, sind nicht konsistent. Achten Sie darauf, welche Namen Sie mit welchen Parametern in Ihrer Konfigurationsdatei und KAFKA_OPTS-Umgebungsvariablen verwenden.

Weitere Informationen zum Zugriff auf Ihre ZooKeeper Apache-Knoten mit TLS finden Sie unter KIP-515: Aktivieren Sie den ZK-Client, um die neue TLS-unterstützte Authentifizierung zu verwenden.

Protokollierung

Sie können Apache Kafka-Broker-Protokolle an einen oder mehrere der folgenden Zieltypen senden: Amazon CloudWatch Logs, Amazon S3, Amazon Data Firehose. Sie können Amazon MSK API-Aufrufe auch mit AWS CloudTrail protokollieren.

Broker-Protokolle

Broker-Protokolle ermöglichen es Ihnen, Probleme mit Ihren Apache-Kafka-Anwendungen zu beheben und die Kommunikation der Anwendungen mit Ihrem MSK-Cluster zu analysieren. Sie können Ihren neuen oder vorhandenen MSK-Cluster so konfigurieren, dass Brokerprotokolle auf INFO-Ebene an eine oder mehrere der folgenden Arten von Zielressourcen gesendet werden: eine CloudWatch Protokollgruppe, ein S3-Bucket, ein Firehose-Lieferstream. Über Firehose können Sie dann die Protokolldaten aus Ihrem Lieferstream an den OpenSearch Service übermitteln. Sie müssen

eine Zielressource erstellen, bevor Sie Ihren Cluster so konfigurieren, dass er Broker-Protokolle dahin übermittelt. Amazon MSK erstellt diese Zielressourcen nicht für Sie, wenn sie nicht bereits vorhanden sind. Informationen zu diesen drei Arten von Zielressourcen und deren Erstellung finden Sie in der folgenden Dokumentation:

- CloudWatch Amazon-Protokolle
- Amazon S3
- Amazon Data Firehose

Erforderliche Berechtigungen

Um ein Ziel für Amazon-MSK-Broker-Protokolle zu konfigurieren, muss die IAM-Identität, die Sie für Amazon-MSK-Aktionen verwenden, über die in der Richtlinie AWS verwaltete Richtlinie: AmazonMSK FullAccess beschriebenen Berechtigungen verfügen.

Um Broker-Protokolle an einen S3-Bucket zu streamen, benötigen Sie auch die Berechtigung `s3:PutBucketPolicy`. Informationen zu S3-Bucket-Richtlinien finden Sie unter Wie füge ich eine S3-Bucket-Richtlinie hinzu? im Amazon-S3-Benutzerhandbuch. Informationen zu IAM-Richtlinien im Allgemeinen finden Sie unter Zugriffsverwaltung im IAM-Benutzerhandbuch.

Erforderliche KMS-Schlüsselrichtlinie zur Verwendung mit SSE-KMS-Buckets

Wenn Sie die serverseitige Verschlüsselung für Ihren S3-Bucket mithilfe von AWS KMS verwalteten Schlüsseln (SSE-KMS) mit einem vom Kunden verwalteten Schlüssel aktiviert haben, fügen Sie der Schlüsselrichtlinie für Ihren KMS-Schlüssel Folgendes hinzu, damit Amazon MSK Brokerdateien in den Bucket schreiben kann.

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

Konfiguration von Broker-Protokollen mit dem AWS Management Console

Wenn Sie einen neuen Cluster erstellen, suchen Sie im Abschnitt Überwachung nach der Überschrift Bereitstellung von Broker-Protokollen. Sie können die Ziele angeben, an die Amazon MSK die Broker-Protokolle bereitstellen soll.

Wählen Sie für einen vorhandenen Cluster den Cluster aus der Cluster-Liste aus und wählen Sie dann die Registerkarte Eigenschaften. Scrollen Sie nach unten zum Abschnitt Protokoll-Bereitstellung und wählen Sie dann die Schaltfläche Bearbeiten. Sie können die Ziele angeben, an die Amazon MSK die Broker-Protokolle bereitstellen soll.

Konfiguration von Brokerprotokollen mit dem AWS CLI

Wenn Sie die Befehle `create-cluster` oder `update-monitoring` verwenden, können Sie optional den Parameter `logging-info` angeben und eine JSON-Struktur wie im folgenden Beispiel an ihn übergeben. In diesem JSON sind alle drei Zieltypen optional.

```
{
  "BrokerLogs": {
    "S3": {
      "Bucket": "ExampleBucketName",
      "Prefix": "ExamplePrefix",
      "Enabled": true
    },
    "Firehose": {
      "DeliveryStream": "ExampleDeliveryStreamName",
      "Enabled": true
    },
    "CloudWatchLogs": {
      "Enabled": true,
      "LogGroup": "ExampleLogGroupName"
    }
  }
}
```

Konfigurieren von Broker-Protokollen mithilfe der API

Sie können die optionale `loggingInfo` Struktur in der JSON-Datei angeben, die Sie an die CreateClusterUpdateMonitoringOR-Operationen übergeben.

> **Note**
>
> Wenn die Broker-Protokollierung aktiviert ist, protokolliert Amazon MSK standardmäßig Protokolle auf `INFO`-Ebene an die angegebenen Ziele. Benutzer von Apache Kafka 2.4.X und höher können jedoch die Broker-Protokollierungsebene jedoch dynamisch auf eine der log4j-Protokollierungsebenen festlegen. Informationen zur dynamischen Festlegung der Broker-Protokollierungsebene finden Sie unter KIP-412: Erweitern der Admin-API zur Unterstützung dynamischer Anwendungs-Protokollierungsebenen. Wenn Sie die Protokollebene dynamisch auf `DEBUG` oder setzen`TRACE`, empfehlen wir, Amazon S3 oder Firehose als Protokollziel zu verwenden. Wenn Sie CloudWatch Logs als Protokollziel verwenden und die `TRACE` Protokollierung dynamisch aktivieren `DEBUG` oder abgleichen, kann Amazon MSK kontinuierlich eine Stichprobe von Protokollen bereitstellen. Dies kann die Leistung des Brokers erheblich beeinträchtigen und sollte nur verwendet werden, wenn die `INFO`-Protokollierungsebene nicht ausführlich genug ist, um die Grundursache eines Problems zu ermitteln.

Protokollierung von AWS CloudTrail-API-Aufrufen mit

> **Note**
>
> AWS CloudTrail Protokolle sind für Amazon MSK nur verfügbar, wenn Sie sie verwendenIAM-Zugriffssteuerung.

Amazon MSK ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Service in Amazon MSK ausgeführt wurden. CloudTrail erfasst API-Aufrufe als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der Amazon-MSK-Konsole und Code-Aufrufe an die Amazon-MSK-API-Vorgänge. Es werden auch Apache-Kafka-Aktionen wie das Erstellen und Ändern von Themen und Gruppen erfasst.

Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Übermittlung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für Amazon

MSK. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Amazon MSK oder die Apache Kafka-Aktion gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Weitere Informationen darüber CloudTrail, einschließlich der Konfiguration und Aktivierung, finden Sie im AWS CloudTrail Benutzerhandbuch.

Amazon MSK-Informationen in CloudTrail

CloudTrail ist in Ihrem Amazon Web Services Services-Konto aktiviert, wenn Sie das Konto erstellen. Wenn unterstützte Ereignisaktivitäten in einem MSK-Cluster auftreten, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen im CloudTrail Ereignisverlauf in einem Ereignis aufgezeichnet. Sie können die neusten Ereignisse in Ihrem Amazon Web Services-Konto anzeigen, suchen und herunterladen. Weitere Informationen finden Sie unter Ereignisse mit CloudTrail Ereignisverlauf anzeigen.

Zur kontinuierlichen Aufzeichnung von Ereignissen in Ihrem Amazon-Web-Services-Konto, einschließlich Ereignissen für Amazon MSK, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle -Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS -Partition und stellt die Protokolldateien in dem von Ihnen angegebenen Amazon-S3-Bucket bereit. Darüber hinaus können Sie andere Amazon-Dienste so konfigurieren, dass sie die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter analysieren und darauf reagieren. Weitere Informationen finden Sie hier:

- Übersicht zum Erstellen eines Trails
- CloudTrail Unterstützte Dienste und Integrationen
- Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

Amazon MSK protokolliert alle Amazon MSK-Operationen als Ereignisse in CloudTrail Protokolldateien. Darüber hinaus protokolliert es die folgenden Apache-Kafka-Aktionen.

- Kafka-Cluster: DescribeClusterDynamicConfiguration
- Kafka-Cluster: AlterClusterDynamicConfiguration

- Kafka-Cluster: CreateTopic
- Kafka-Cluster: DescribeTopicDynamicConfiguration
- Kafka-Cluster: AlterTopic
- Kafka-Cluster: AlterTopicDynamicConfiguration
- Kafka-Cluster: DeleteTopic

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Root-Benutzer- oder AWS Identity and Access Management (IAM-) Benutzeranmeldedaten gestellt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie unter dem CloudTrail UserIdentity-Element.

Beispiel: Einträge in der Amazon-MSK-Protokolldatei

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält unter anderem Informationen über die angeforderte Aktion, das Datum und die Uhrzeit der Aktion sowie über die Anforderungsparameter. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe und Apache Kafka-Aktionen, sie erscheinen also nicht in einer bestimmten Reihenfolge.

Das folgende Beispiel zeigt CloudTrail Protokolleinträge, die die Aktionen `DescribeCluster` und `DeleteCluster` Amazon MSK demonstrieren.

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

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `kafka-cluster:CreateTopic` Aktion demonstriert.

```
{
  "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"
}
```

Compliance-Validierung für Amazon Managed Streaming für Apache Kafka

Externe Prüfer bewerten im Rahmen verschiedener AWS -Compliance-Programme die Sicherheit und Compliance von Amazon Managed Streaming für Apache Kafka. Dazu gehören PCI und HIPAA BAA.

Eine Liste der AWS Services im Rahmen bestimmter Compliance-Programme finden Sie unter Amazon Services in Umfang nach Compliance-Programm . Allgemeine Informationen finden Sie unter AWS Compliance-Programme AWS .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

Ihre Compliance-Verantwortung bei der Nutzung von Amazon MSK hängt von der Sensibilität Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS bietet die folgenden Ressourcen zur Unterstützung bei der Einhaltung von Vorschriften:

- Schnellstartanleitungen für Sicherheit und Compliance – In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Bereitstellung von sicherheits- und konformitätsorientierten Basisumgebungen auf AWS angegeben.
- Whitepaper „Architecting for HIPAA Security and Compliance“ — In diesem Whitepaper wird beschrieben, wie Unternehmen HIPAA-konforme Anwendungen erstellen können AWS .
- AWS Compliance-Ressourcen — Diese Sammlung von Arbeitsmappen und Leitfäden kann auf Ihre Branche und Ihren Standort zutreffen.
- Bewertung von Ressourcen anhand von Regeln im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub— Dieser AWS Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus, sodass Sie überprüfen können AWS , ob Sie die Sicherheitsstandards und Best Practices der Branche einhalten.

Ausfallsicherheit in Amazon Managed Streaming für Apache Kafka

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones. AWS Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Infrastruktursicherheit in Amazon Managed Streaming für Apache Kafka

Als verwalteter Service ist Amazon Managed Streaming for Apache Kafka durch die AWS globalen Netzwerksicherheitsverfahren geschützt, die im Whitepaper Amazon Web Services: Sicherheitsprozesse im Überblick beschrieben werden.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Amazon MSK zuzugreifen. Kunden müssen Transport Layer Security (TLS) 1.0 oder neuer unterstützen. Wir empfehlen TLS 1.2 oder höher. Clients müssen außerdem Verschlüsselungssammlungen mit PFS (Perfect Forward Secrecy) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) unterstützen. Die meisten modernen Systemen wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Herstellen einer Verbindung mit einem Amazon-MSK-Cluster

Standardmäßig können Clients nur dann auf einen MSK-Cluster zugreifen, wenn sie sich in derselben VPC wie der Cluster befinden. Die gesamte Kommunikation zwischen Ihren Kafka-Clients und Ihrem MSK-Cluster ist standardmäßig privat und Ihre Streaming-Daten werden niemals über das Internet übertragen. Damit Sie von einem Client, der sich in derselben VPC wie der Cluster befindet, eine Verbindung zu Ihrem MSK-Cluster herstellen können, benötigt die Sicherheitsgruppe des Clusters eine eingehende Regel, die Datenverkehr von der Sicherheitsgruppe des Clients akzeptiert. Informationen zum Einrichten dieser Regeln finden Sie unter Sicherheitsgruppenregeln. Ein Beispiel für den Zugriff auf einen Cluster von einer Amazon-EC2-Instance aus, die sich in derselben VPC wie der Cluster befindet, finden Sie unter Erste Schritte.

Informationen zum Herstellen einer Verbindung mit Ihrem MSK-Cluster von einem Client aus, der sich außerhalb der Cluster-VPC befindet, finden Sie unter Zugriff von innerhalb, AWS aber außerhalb der Cluster-VPC.

Themen

Öffentlicher Zugriff

Amazon MSK bietet Ihnen die Möglichkeit, den öffentlichen Zugriff auf die Broker von MSK-Clustern zu aktivieren, auf denen Apache Kafka 2.6.0 oder spätere Versionen ausgeführt werden. Aus Sicherheitsgründen können Sie den öffentlichen Zugriff nicht aktivieren, während Sie einen MSK-Cluster erstellen. Sie können jedoch einen vorhandenen Cluster aktualisieren, um ihn öffentlich zugänglich zu machen. Sie können auch einen neuen Cluster erstellen und ihn dann aktualisieren, um ihn öffentlich zugänglich zu machen.

Sie können den öffentlichen Zugriff auf einen MSK-Cluster ohne zusätzliche Kosten aktivieren. Für die AWS Datenübertragung innerhalb und aus dem Cluster fallen jedoch die Standardkosten für die Datenübertragung an. Weitere Informationen den Preisgestaltung finden Sie unter On-Demand-Preise von Amazon EC2.

Um den öffentlichen Zugriff auf einen Cluster zu aktivieren, stellen Sie zunächst sicher, dass der Cluster alle der folgenden Bedingungen erfüllt:

- Die Subnetze, die dem Cluster zugeordnet sind, müssen öffentlich sein. Das bedeutet, dass den Subnetzen eine Routing-Tabelle mit einem angeschlossenen Internet-Gateway zugeordnet sein muss. Weitere Informationen zum Erstellen und Anfügen eines Internet-Gateways finden Sie unter Internet-Gateways im Amazon-VPC-Benutzerhandbuch.
- Die nicht authentifizierte Zugriffssteuerung muss ausgeschaltet sein und mindestens eine der folgenden Zugriffssteuerungs-Methoden muss aktiviert sein: SASL/IAM, SASL/SCRAM, mTLS. Weitere Informationen zum Aktualisieren der Zugriffssteuerungs-Methode eines Clusters finden Sie unter the section called “Aktualisieren der Sicherheit”.
- Die Verschlüsselung innerhalb des Clusters muss aktiviert sein. Die Einstellung Ein ist die Standardeinstellung beim Erstellen eines Clusters. Es ist nicht möglich, die Verschlüsselung innerhalb des Clusters für einen Cluster zu aktivieren, der mit ausgeschalteter Verschlüsselung erstellt wurde. Es ist daher nicht möglich, den öffentlichen Zugriff für einen Cluster zu aktivieren, der mit deaktivierter Verschlüsselung erstellt wurde.
- Der Klartext-Datenverkehr zwischen Brokern und Clients muss Aus sein. Informationen darüber, wie Sie ihn ausschalten können, wenn er eingeschaltet ist, finden Sie unter the section called “Aktualisieren der Sicherheit”.
- Wenn Sie die Zugriffssteuerungs-Methoden SASL/SCRAM oder mTLS verwenden, müssen Sie Apache-Kafka-ACLs für Ihren Cluster festlegen. Nachdem Sie die Apache-Kafka-ACLs für Ihren Cluster festgelegt haben, aktualisieren Sie die Cluster-Konfiguration, sodass die Eigenschaft `allow.everyone.if.no.acl.found` für den Cluster auf Falsch gesetzt wird. Weitere Informationen zum Aktualisieren der Konfiguration eines Clusters finden Sie unter the section called “Konfigurationsvorgänge”. Wenn Sie IAM-Zugriffssteuerung verwenden und Autorisierungsrichtlinien anwenden oder Ihre Autorisierungsrichtlinien aktualisieren möchten, finden Sie weitere Informationen unter the section called “IAM-Zugriffssteuerung”. Informationen zu Apache-Kafka-ACLs finden Sie unter the section called “Apache Kafka ACLs”.

Nachdem Sie sichergestellt haben, dass ein MSK-Cluster die oben aufgeführten Bedingungen erfüllt, können Sie die AWS Management Console AWS CLI, oder die Amazon MSK-API verwenden, um den öffentlichen Zugriff zu aktivieren. Nachdem Sie den öffentlichen Zugriff auf einen Cluster aktiviert haben, können Sie eine öffentliche Bootstrap-Broker-Zeichenfolge für diesen Cluster abrufen. Weitere Informationen zum Abrufen der Bootstrap-Broker für einen Cluster finden Sie unter the section called “Abrufen der Bootstrap-Broker”.

> **Important**
>
> Stellen Sie neben der Aktivierung des öffentlichen Zugriffs sicher, dass die Sicherheitsgruppen des Clusters über TCP-Regeln für eingehenden Datenverkehr verfügen, die öffentlichen Zugriff von Ihrer IP-Adresse aus ermöglichen. Wir empfehlen, dass Sie diese Regeln so restriktiv wie möglich gestalten. Weitere Informationen zu Sicherheitsgruppen und Regeln für eingehenden Datenverkehr finden Sie unter Sicherheitsgruppen für Ihre VPC im Amazon-VPC-Benutzerhandbuch. Portnummern finden Sie unter the section called “Port-Informationen”. Anweisungen zum Ändern der Sicherheitsgruppe eines Clusters finden Sie unter the section called “Ändern von Sicherheitsgruppen”.

> **Note**
>
> Wenn Sie die folgenden Anweisungen verwenden, um den öffentlichen Zugriff zu aktivieren und dann immer noch nicht auf den Cluster zugreifen können, finden Sie dazu Informationen unter the section called “Es kann nicht auf einen Cluster zugegriffen werden, für den der öffentliche Zugriff aktiviert ist”.

Aktivieren des öffentlichen Zugriffs mit der Konsole

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon MSK-Konsole unter https://console.aws.amazon.com/msk/home?region=us-east-1#/home/.
2. Wählen Sie in der Cluster-Liste den Cluster aus, für den Sie den öffentlichen Zugriff aktivieren möchten.
3. Wählen Sie die Registerkarte Eigenschaften und suchen Sie dann den Abschnitt Netzwerkeinstellungen.
4. Wählen Sie Öffentlichen Zugriff bearbeiten.

Aktivieren Sie den öffentlichen Zugriff mit dem AWS CLI

1. Führen Sie den folgenden AWS CLI Befehl aus *ClusterArn*und ersetzen Sie *Current-Cluster-Version durch den ARN und die aktuelle Version des Clusters*. Verwenden Sie den Befehl DescribeClusteroperation oder describe-cluster, um die aktuelle Version des Clusters zu ermitteln. AWS CLI `KTVPDKIKX0DER` ist ein Beispiel für eine Version.

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

Die Ausgabe dieses `update-connectivity`-Befehls sieht wie das folgende JSON-Beispiel aus.

```
{
    "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**
>
> Um den öffentlichen Zugriff zu deaktivieren, verwenden Sie einen ähnlichen AWS CLI Befehl, jedoch mit den folgenden Verbindungsinformationen:
>
> ```
> '{"PublicAccess": {"Type": "DISABLED"}}'
> ```

2. Um das Ergebnis des `update-connectivity` Vorgangs zu erhalten, führen Sie den folgenden Befehl aus und ersetzen Sie *`ClusterOperationArn`* durch den ARN, den Sie in der Ausgabe des `update-connectivity` Befehls erhalten haben.

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

Die Ausgabe dieses `describe-cluster-operation`-Befehls sieht wie das folgende JSON-Beispiel aus.

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

Wenn `OperationState` den Wert „`UPDATE_IN_PROGRESS`“ aufweist, warten Sie eine Weile, bevor Sie den `describe-cluster-operation`-Befehl erneut ausführen.

Aktivieren des öffentlichen Zugriffs mithilfe der Amazon-MSK-API

- Informationen zum Aktivieren oder Deaktivieren des öffentlichen Zugriffs auf einen Cluster mithilfe der API finden Sie unter UpdateConnectivity.

> **Note**
>
> Aus Sicherheitsgründen erlaubt Amazon MSK keinen öffentlichen Zugriff auf Apache ZooKeeper - oder Kraft-Controllerknoten.

Zugriff von innerhalb AWS , aber außerhalb der Cluster-VPC

Um von innerhalb, AWS aber außerhalb der Amazon VPC des Clusters eine Verbindung zu einem MSK-Cluster herzustellen, gibt es die folgenden Optionen.

Amazon-VPC-Peering

Damit Sie von einer VPC aus, die sich von der VPC des Clusters unterscheidet, auf Ihren MSK-Cluster zugreifen können, können Sie eine Peering-Verbindung zwischen den beiden VPCs erstellen. Informationen zum VPC-Peering finden Sie im Amazon VPC-Peering-Handbuch.

AWS Direct Connect

AWS Direct Connect verbindet Ihr lokales Netzwerk AWS über ein standardmäßiges 1-Gigabit- oder 10-Gigabit-Ethernet-Glasfaserkabel. Ein Ende des Kabels ist mit Ihrem Router verbunden, das andere mit einem Router. AWS Direct Connect Mit dieser Verbindung können Sie virtuelle Schnittstellen direkt zur AWS Cloud und Amazon VPC erstellen und dabei Internetdienstanbieter in Ihrem Netzwerkpfad umgehen. Weitere Informationen finden Sie unter AWS Direct Connect.

AWS Transit Gateway

AWS Transit Gateway ist ein Service, mit dem Sie Ihre VPCs und Ihre lokalen Netzwerke mit einem einzigen Gateway verbinden können. Weitere Informationen zur Verwendung von AWS Transit Gateway finden Sie unter AWS Transit Gateway.

VPN-Verbindungen

Sie können die VPC Ihres MSK-Clusters mithilfe der im folgenden Thema beschriebenen VPC-Konnektivitätsoptionen mit Remote-Netzwerken und -Benutzern verbinden: VPN-Verbindungen.

REST-Proxys

Sie können einen REST-Proxy auf einer Instance installieren, die in der Amazon VPC Ihres Clusters ausgeführt wird. Mit REST-Proxys können Ihre Produzenten und Verbraucher über HTTP-API-Anforderungen mit dem Cluster kommunizieren.

Multi-VPC-Konnektivität in mehreren Regionen

Im folgenden Dokument werden Konnektivitätsoptionen für mehrfache VPCs beschrieben, die sich in verschiedenen Regionen befinden: Multi-VPC-Konnektivität in mehreren Regionen.

Private Multi-VPC-Konnektivität in einer einzelnen Region

Private Multi-VPC-Konnektivität (unterstützt von AWS PrivateLink) für Amazon Managed Streaming for Apache Kafka (Amazon MSK) -Cluster ist eine Funktion, mit der Sie Kafka-Clients, die in

verschiedenen Virtual Private Clouds (VPCs) und AWS Konten gehostet werden, schneller mit einem Amazon MSK-Cluster verbinden können.

Weitere Informationen finden Sie unter Multi-VPC-Konnektivität in einer einzelnen Region für kontoübergreifende Kunden.

EC2-Classic-Netzwerke wurden eingestellt

Amazon MSK unterstützt keine Amazon EC2 EC2-Instances mehr, die mit Amazon EC2-Classic-Netzwerken ausgeführt werden.

Weitere Informationen finden Sie unter EC2-Classic Networking wird eingestellt — So bereiten Sie sich darauf vor.

Private Multi-VPC-Konnektivität von Amazon MSK in einer einzelnen Region

Private Multi-VPC-Konnektivität (unterstützt von AWS PrivateLink) für Amazon Managed Streaming for Apache Kafka (Amazon MSK) -Cluster ist eine Funktion, mit der Sie Kafka-Clients, die in verschiedenen Virtual Private Clouds (VPCs) und AWS Konten gehostet werden, schneller mit einem Amazon MSK-Cluster verbinden können.

Private Multi-VPC-Konnektivität ist eine verwaltete Lösung, die die Netzwerkinfrastruktur für Multi-VPC- und kontenübergreifende Konnektivität vereinfacht. Clients können eine Verbindung zum Amazon MSK-Cluster herstellen PrivateLink und gleichzeitig den gesamten Datenverkehr im AWS Netzwerk behalten. Private Multi-VPC-Konnektivität für Amazon MSK-Cluster ist in allen AWS Regionen verfügbar, in denen Amazon MSK verfügbar ist.

Themen

- Was ist private Multi-VPC-Konnektivität?
- Vorteile der privaten Multi-VPC-Konnektivität
- Anforderungen und Einschränkungen für private Multi-VPC-Konnektivität
- Erste Schritte mit privater Multi-VPC-Konnektivität
- Die Autorisierungsschema auf einem Cluster aktualisieren
- Eine verwaltete VPC-Verbindung zu einem Amazon-MSK-Cluster ablehnen
- Eine verwaltete VPC-Verbindung zu einem Amazon-MSK-Cluster löschen
- Berechtigungen für private Multi-VPC-Konnektivität

Was ist private Multi-VPC-Konnektivität?

Private Multi-VPC-Konnektivität für Amazon MSK ist eine Konnektivitätsoption, mit der Sie Apache Kafka-Clients, die in verschiedenen Virtual Private Clouds (VPCs) und AWS Konten gehostet werden, mit einem MSK-Cluster verbinden können.

Amazon MSK vereinfacht den kontoübergreifenden Zugriff mit Cluster-Richtlinien. Diese Richtlinien ermöglichen es dem Clusterbesitzer, anderen AWS Konten Berechtigungen zu erteilen, um eine private Konnektivität zum MSK-Cluster herzustellen.

Vorteile der privaten Multi-VPC-Konnektivität

Private Multi-VPC-Konnektivität bietet mehrere Vorteile gegenüber anderen Konnektivitätslösungen:

- Es automatisiert das Betriebsmanagement der AWS PrivateLink Konnektivitätslösung.
- Es ermöglicht überlappende IPs zwischen verbindenden VPCs, wodurch die Notwendigkeit entfällt, überlappungsfreie IPs, komplexes Peering und Routing-Tabellen zu verwalten, die mit anderen VPC-Konnektivitätslösungen verbunden sind.

Sie verwenden eine Clusterrichtlinie für Ihren MSK-Cluster, um zu definieren, welche AWS Konten berechtigt sind, kontenübergreifende private Konnektivität zu Ihrem MSK-Cluster einzurichten. Der kontoübergreifende Administrator kann Berechtigungen an entsprechende Rollen oder Benutzer delegieren. Bei Verwendung mit der IAM-Client-Authentifizierung können Sie die Cluster-Richtlinie auch verwenden, um die Kafka-Datenebenen-Berechtigungen für die verbindenden Clients detailliert zu definieren.

Anforderungen und Einschränkungen für private Multi-VPC-Konnektivität

Beachten Sie die folgenden MSK-Cluster-Anforderungen für die Ausführung von privater Multi-VPC-Konnektivität:

- Private Multi-VPC-Konnektivität wird nur auf Apache Kafka 2.7.1 oder höher unterstützt. Stellen Sie sicher, dass auf allen Clients, die Sie mit dem MSK-Cluster verwenden, Apache-Kafka-Versionen ausgeführt werden, die mit dem Cluster kompatibel sind.
- Private Multi-VPC-Konnektivität unterstützt die Authentifizierungstypen IAM, TLS und SASL/SCRAM. Nicht authentifizierte Cluster können keine private Multi-VPC-Konnektivität verwenden.
- Wenn Sie die Zugriffssteuerungs-Methoden SASL/SCRAM oder mTLS verwenden, müssen Sie Apache-Kafka-ACLs für Ihren Cluster einrichten. Stellen Sie zunächst die Apache-Kafka-ACLs für Ihren Cluster ein. Aktualisieren Sie anschließend die Konfiguration des Clusters, sodass

die Eigenschaft `allow.everyone.if.no.acl.found` für den Cluster auf Falsch gesetzt ist. Weitere Informationen zum Aktualisieren der Konfiguration eines Clusters finden Sie unter the section called “Konfigurationsvorgänge”. Wenn Sie IAM-Zugriffssteuerung verwenden und Autorisierungsrichtlinien anwenden oder Ihre Autorisierungsrichtlinien aktualisieren möchten, finden Sie weitere Informationen unter the section called “IAM-Zugriffssteuerung”. Informationen zu Apache-Kafka-ACLs finden Sie unter the section called “Apache Kafka ACLs”.

- Private Multi-VPC-Konnektivität unterstützt den Instance-Typ t3.small nicht.
- Private Multi-VPC-Konnektivität wird nicht regionsübergreifend unterstützt, sondern nur AWS für Konten innerhalb derselben AWS Region.
- Amazon MSK unterstützt keine private Multi-VPC-Konnektivität zu ZooKeeper-Knoten.

Erste Schritte mit privater Multi-VPC-Konnektivität

Themen

- Schritt 1: Auf dem MSK-Cluster in Konto A die Multi-VPC-Konnektivität für das IAM-Authentifizierungsschema auf dem Cluster aktivieren
- Schritt 2: Eine Cluster-Richtlinie an den MSK-Cluster anhängen
- Schritt 3: Kontoübergreifende Benutzeraktionen zur Konfiguration von clientverwalteten VPC-Verbindungen

In diesem Tutorial wird ein gängiger Anwendungsfall als Beispiel dafür verwendet, wie Sie Multi-VPC-Konnektivität verwenden können, um einen Apache Kafka-Client privat mit einem MSK-Cluster von innerhalb AWS, aber außerhalb der VPC des Clusters zu verbinden. Für diesen Prozess muss der kontoübergreifende Benutzer eine MSK-verwaltete VPC-Verbindung und -Konfiguration für jeden Client erstellen, einschließlich der erforderlichen Client-Berechtigungen. Der Prozess erfordert außerdem, dass der Eigentümer des MSK-Clusters die PrivateLink Konnektivität auf dem MSK-Cluster aktiviert und Authentifizierungsschemata zur Steuerung des Zugriffs auf den Cluster auswählt.

In verschiedenen Teilen dieses Tutorials wählen wir Optionen aus, die für dieses Beispiel gelten. Dies bedeutet nicht, dass dies die einzigen Optionen sind, um einen MSK-Cluster oder Client-Instances einzurichten.

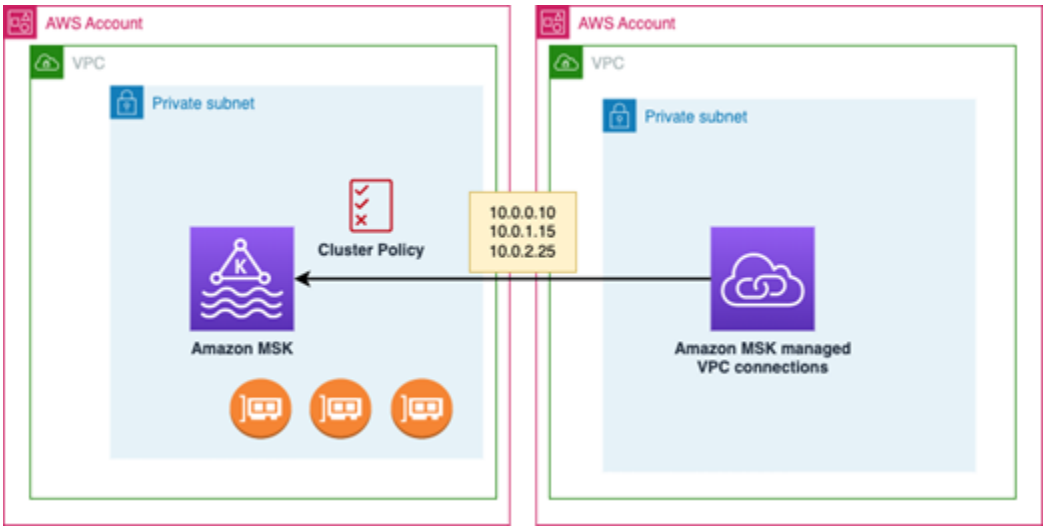
Die Netzwerkkonfiguration für diesen Anwendungsfall lautet wie folgt:

- Ein kontoübergreifender Benutzer (Kafka-Client) und ein MSK-Cluster befinden sich in demselben/derselben AWS -Netzwerk/-Region, aber in unterschiedlichen Konten:

  - MSK-Cluster in Konto A
  - Kafka-Client in Konto B
- Der kontoübergreifende Benutzer stellt mithilfe des IAM-Authentifizierungsschemas eine private Verbindung zum MSK-Cluster her.

In diesem Tutorial wird davon ausgegangen, dass es einen bereitgestellten MSK-Cluster gibt, der mit Apache Kafka Version 2.7.1 oder höher erstellt wurde. Der MSK-Cluster muss sich im ACTIVE-Status befinden, bevor Sie mit dem Konfigurationsprozess beginnen können. Um potenziellen Datenverlust oder Ausfallzeiten zu vermeiden, sollten Clients, die eine private Multi-VPC-Verbindung nutzen, um eine Verbindung zum Cluster herzustellen, Apache-Kafka-Versionen verwenden, die mit dem Cluster kompatibel sind.

Das folgende Diagramm zeigt die Architektur der Amazon MSK Multi-VPC-Konnektivität, die mit einem Client in einem anderen Konto verbunden ist. AWS



Schritt 1: Auf dem MSK-Cluster in Konto A die Multi-VPC-Konnektivität für das IAM-Authentifizierungsschema auf dem Cluster aktivieren

Der MSK-Cluster-Besitzer muss die Konfigurationseinstellungen für den MSK-Cluster vornehmen, nachdem der Cluster erstellt wurde und sich im Status ACTIVE befindet.

Der Cluster-Besitzer aktiviert private Multi-VPC-Konnektivität auf dem ACTIVE-Cluster für alle Authentifizierungsschemas, die auf dem Cluster aktiv sein werden. Dies kann mithilfe der UpdateSecurity API - oder MSK-Konsole erfolgen. Die Authentifizierungsschema IAM, SASL/SCRAM und TLS unterstützen private Multi-VPC-Konnektivität. Private Multi-VPC-Konnektivität kann für nicht authentifizierte Cluster nicht aktiviert werden.

Für diesen Anwendungsfall konfigurieren Sie den Cluster für die Verwendung des IAM-Authentifizierungsschemas.

> **Note**
>
> Wenn Sie Ihren MSK-Cluster für die Verwendung des SASL/SCRAM-Authentifizierungsschemas konfigurieren, ist die Apache-Kafka-ACLs-Eigenschaft „`allow.everyone.if.no.acl.found=false`“ obligatorisch. Siehe Apache-Kafka-ACLs.

Wenn Sie die privaten Multi-VPC-Konnektivitätseinstellungen aktualisieren, startet Amazon MSK einen fortlaufenden Neustart der Broker-Knoten, um die Broker-Konfigurationen zu aktualisieren. Dieser Vorgang kann bis zu 30 Minuten dauern. Sie können keine weiteren Aktualisierungen am Cluster vornehmen, während die Konnektivität aktualisiert wird.

Aktivieren der Multi-VPC für ausgewählte Authentifizierungsschemas auf dem Cluster in Konto A mithilfe der Konsole

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/ für das Konto, in dem sich der Cluster befindet.
2. Wählen Sie im Navigationsbereich unter MSK-Cluster die Option Cluster aus, um die Liste der Cluster im Konto anzuzeigen.
3. Wählen Sie den Cluster aus, der für private Multi-VPC-Konnektivität konfiguriert werden soll. Der Cluster muss sich im ACTIVE-Status befinden.
4. Wählen Sie die Eigenschaften-Registerkarte des Clusters und wechseln Sie dann zu den Netzwerk-Einstellungen.
5. Wählen Sie das Dropdown-Menü Bearbeiten und dann Multi-VPC-Konnektivität aktivieren.
6. Wählen Sie einen oder mehrere Authentifizierungstypen aus, die Sie für diesen Cluster aktivieren möchten. Wählen Sie für diesen Anwendungsfall die IAM-rollenbasierte Authentifizierung.
7. Wählen Sie Änderungen speichern aus.

Example - UpdateConnectivity API, die Authentifizierungsschemata für private Verbindungen mit mehreren VPC auf einem Cluster aktiviert

Als Alternative zur MSK-Konsole können Sie die UpdateConnectivity API verwenden, um private Multi-VPC-Konnektivität zu aktivieren und Authentifizierungsschemata auf einem ACTIVE-Cluster zu

konfigurieren. Das folgende Beispiel zeigt, dass das IAM-Authentifizierungsschema für den Cluster aktiviert ist.

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

Amazon MSK erstellt die Netzwerkinfrastruktur, die für private Konnektivität erforderlich ist. Amazon MSK erstellt außerdem einen neuen Satz von Bootstrap-Broker-Endpunkten für jeden Authentifizierungstyp, der private Konnektivität erfordert. Beachten Sie, dass das Klartext-Authentifizierungsschema keine private Multi-VPC-Konnektivität unterstützt.

Schritt 2: Eine Cluster-Richtlinie an den MSK-Cluster anhängen

Der Cluster-Besitzer kann eine Cluster-Richtlinie (auch als ressourcenbasierte Richtlinie bezeichnet) an den MSK-Cluster anhängen, in dem Sie die private Multi-VPC-Konnektivität aktivieren. Die Cluster-Richtlinie erteilt den Clients die Berechtigung, von einem anderen Konto aus auf den Cluster zuzugreifen. Bevor Sie die Cluster-Richtlinie bearbeiten können, benötigen Sie die Konto-ID(s) für die Konten, die berechtigt sein sollen, auf den MSK-Cluster zuzugreifen. Siehe Funktionsweise von Amazon MKS mit IAM.

Der Cluster-Besitzer muss dem MSK-Cluster eine Cluster-Richtlinie hinzufügen, die den kontoübergreifenden Benutzer in Konto B autorisiert, Bootstrap-Broker für den Cluster abzurufen und die folgenden Aktionen auf dem MSK-Cluster in Konto A zu autorisieren:

- CreateVpcVerbindung
- GetBootstrapMakler
- DescribeCluster
- DescribeClusterV2

Example

Als Referenz finden Sie im Folgenden ein JSON-Beispiel für eine grundlegende Cluster-Richtlinie, ähnlich der Standardrichtlinie, die im IAM-Richtlinien-Editor der MSK-Konsole angezeigt wird.

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

Eine Cluster-Richtlinie an den MSK-Cluster anhängen

1. Wählen Sie in der Amazon-MSK-Konsole unter MSK-Cluster die Option Cluster aus.
2. Scrollen Sie nach unten zu Sicherheitseinstellungen und wählen Sie Cluster-Richtlinie bearbeiten.
3. Wählen Sie in der Konsole auf dem Bildschirm Cluster-Richtlinie bearbeiten die Option Basisrichtlinie für Multi-VPC-Konnektivität.
4. Geben Sie im Feld Konto-ID die Konto-ID für jedes Konto ein, das berechtigt sein soll, auf diesen Cluster zuzugreifen. Wenn Sie die ID eingeben, wird sie automatisch in die angezeigte JSON-Syntax der Richtlinie kopiert. In unserem Beispiel für eine Cluster-Richtlinie lautet die Konto-ID `123456789012`.
5. Wählen Sie Änderungen speichern aus.

Informationen zu APIs für Cluster-Richtlinien finden Sie unter Ressourcenbasierte Amazon-MSK-Richtlinien.

Schritt 3: Kontoübergreifende Benutzeraktionen zur Konfiguration von clientverwalteten VPC-Verbindungen

Um private Multi-VPC-Konnektivität zwischen einem Client in einem anderen Konto als dem MSK-Cluster einzurichten, erstellt der kontoübergreifende Benutzer eine verwaltete VPC-Verbindung für den Client. Durch Wiederholen dieses Verfahrens können mehrere Clients mit dem MSK-Cluster verbunden werden. Für diesen Anwendungsfall konfigurieren Sie nur einen Client.

Clients können die unterstützten Authentifizierungsschema IAM, SASL/SCRAM oder TLS verwenden. Jeder verwalteten VPC-Verbindung kann nur ein Authentifizierungsschema zugeordnet sein. Das Client-Authentifizierungsschema muss auf dem MSK-Cluster konfiguriert werden, zu dem der Client eine Verbindung herstellt.

Für diesen Anwendungsfall konfigurieren Sie das Client-Authentifizierungsschema so, dass der Client in Konto B das IAM-Authentifizierungsschema verwendet.

Voraussetzungen

Dieser Vorgang erfordert die folgenden Elemente:

- Die zuvor erstellte Clusterrichtlinie, die dem Client in Konto B die Berechtigung erteilt, Aktionen auf dem MSK-Cluster in Konto A durchzuführen.
- Eine dem Client in Konto B zugeordnete Identitätsrichtlinie, die Berechtigungen für `kafka:CreateVpcConnectionec2:CreateTags`, `ec2:CreateVPCEndpoint` und `ec2:DescribeVpcAttribute` Aktionen gewährt.

Example

Zum Nachschlagen finden Sie nachstehend ein JSON-Beispiel für eine grundlegende Client-Identitätsrichtlinie.

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

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

So erstellen Sie eine verwaltete VPC-Verbindung für einen Client in Konto B

1. Rufen Sie vom Cluster-Administrator den Cluster-ARN des MSK-Clusters in Konto A ab, zu dem der Client in Konto B eine Verbindung herstellen soll. Notieren Sie sich den Cluster-ARN, um ihn später zu verwenden.
2. Wählen Sie in der MSK-Konsole für das Client-Konto B Verwaltete VPC-Verbindungen und dann Verbindung erstellen.
3. Fügen Sie im Bereich Verbindungseinstellungen den Cluster-ARN in das Cluster-ARN-Textfeld ein, und wählen Sie dann Überprüfen.
4. Wählen Sie den Authentifizierungstyp für den Client in Konto B. Wählen Sie für diesen Anwendungsfall IAM, wenn Sie die Client-VPC-Verbindung erstellen.
5. Wählen Sie die VPC für den Client aus.
6. Wählen Sie mindestens zwei Availability Zones und zugehörige Subnetze. Sie können die Verfügbarkeitszonen-IDs in den Clusterdetails der AWS Management Console oder mithilfe der DescribeClusterAPI oder des AWS CLI-Befehls describe-cluster abrufen. Die Zonen-IDs, die Sie für das Client-Subnetz angeben, müssen mit denen des Cluster-Subnetzes übereinstimmen. Wenn die Werte für ein Subnetz fehlen, erstellen Sie zunächst ein Subnetz mit derselben Zonen-ID wie Ihr MSK-Cluster.
7. Wählen Sie eine Sicherheitsgruppe für diese VPC-Verbindung aus. Sie können die Standardsicherheitsgruppe verwenden. Weitere Informationen zum Konfigurieren einer Sicherheitsgruppe finden Sie unter Steuern des Datenverkehrs zu Ressourcen mithilfe von Sicherheitsgruppen.
8. Wählen Sie Verbindung erstellen.
9. Informationen, um die Liste der neuen Bootstrap-Broker-Zeichenfolgen von der MSK-Konsole des kontoübergreifenden Benutzers abzurufen (Cluster-Details > Verwaltete VPC-Verbindung), finden Sie in den Bootstrap-Broker-Zeichenfolgen unter „Cluster-Verbindungszeichenfolge.“

Vom Kundenkonto B aus kann die Liste der Bootstrap-Broker angezeigt werden, indem Sie die Broker-API aufrufen oder die Liste der GetBootstrapBootstrap-Broker in den Cluster-Details der Konsole aufrufen.

10. Aktualisieren Sie die mit den VPC-Verbindungen verknüpften Sicherheitsgruppen wie folgt:

    a. Legen Sie Regeln für eingehenden Datenverkehr für die PrivateLink VPC fest, um den gesamten Datenverkehr für den IP-Bereich aus dem Konto B-Netzwerk zuzulassen.

    b. [Optional] Legen Sie die Konnektivität für Regeln für ausgehenden Datenverkehr zum MSK-Cluster fest. Wählen Sie die Sicherheitsgruppe in der VPC-Konsole, Regeln für ausgehenden Datenverkehr bearbeiten und fügen Sie eine Regel für benutzerdefinierten TCP-Datenverkehr für die Portbereiche 14001–14100 hinzu. Der Multi-VPC-Network-Load-Balancer überwacht die Portbereiche 14001–14100. Siehe Network Load Balancers.

11. Konfigurieren Sie den Client in Konto B so, dass er die neuen Bootstrap-Broker für private Multi-VPC-Konnektivität verwendet, um eine Verbindung zum MSK-Cluster in Konto A herzustellen. Siehe Daten produzieren und verbrauchen.

Nach Abschluss der Autorisierung erstellt Amazon MSK eine verwaltete VPC-Verbindung für jede angegebene VPC und jedes Authentifizierungsschema. Die gewählte Sicherheitsgruppe ist der jeweiligen Verbindung zugeordnet. Diese verwaltete VPC-Verbindung wird von Amazon MSK so konfiguriert, dass sie sich privat mit den Brokern verbindet. Sie können die neuen Bootstrap-Broker verwenden, um eine private Verbindung zum Amazon-MSK-Cluster herzustellen.

Die Autorisierungsschema auf einem Cluster aktualisieren

Die private Multi-VPC-Konnektivität unterstützt mehrere Authentifizierungsschema: SASL/SCRAM. IAM und TLS. Der Cluster-Besitzer kann die private Konnektivität für ein oder mehrere Authentifizierungsschema ein- und ausschalten. Der Cluster muss sich im Status ACTIVE befinden, um diese Aktion ausführen zu können.

So aktivieren Sie ein Authentifizierungsschema mit der Amazon-MSK-Konsole

1. Öffnen Sie die Amazon-MSK-Konsole unter AWS Management Console für den Cluster, den Sie bearbeiten möchten.
2. Wählen Sie im Navigationsbereich unter MSK-Cluster die Option Cluster aus, um die Liste der Cluster im Konto anzuzeigen.
3. Wählen Sie den Cluster aus, den Sie bearbeiten möchten. Der Cluster muss sich im ACTIVE-Status befinden.

4. Wählen Sie die Registerkarte Eigenschaften des Clusters und wechseln Sie dann zu Netzwerkeinstellungen.
5. Wählen Sie das Dropdown-Menü Bearbeiten und dann Multi-VPC-Konnektivität aktivieren, um ein neues Authentifizierungsschema einzuschalten.
6. Wählen Sie einen oder mehrere Authentifizierungstyp(en) aus, die Sie für diesen Cluster aktivieren möchten.
7. Wählen Sie Auswahl aktivieren.

Wenn Sie ein neues Authentifizierungsschema aktivieren, sollten Sie auch neue verwaltete VPC-Verbindungen für das neue Authentifizierungsschema erstellen und Ihre Clients so aktualisieren, dass sie die für das neue Authentifizierungsschema spezifischen Bootstrap-Broker verwenden.

So deaktivieren Sie ein Authentifizierungsschema mithilfe der Amazon-MSK-Konsole

> **Note**
>
> Wenn Sie private Multi-VPC-Konnektivität für Authentifizierungsschemas deaktivieren, wird die gesamte konnektivitätsbezogene Infrastruktur, einschließlich der verwalteten VPC-Verbindungen, gelöscht.

Wenn Sie private Multi-VPC-Konnektivität für Authentifizierungsschemas deaktivieren, ändern sich bestehende VPC-Verbindungen auf der Client-Seite in INACTIVE, und die PrivateLink-Infrastruktur auf der Cluster-Seite, einschließlich der verwalteten VPC-Verbindungen, wird entfernt. Der kontoübergreifende Benutzer kann nur die inaktive VPC-Verbindung löschen. Wenn die private Konnektivität auf dem Cluster wieder aktiviert wird, muss der kontoübergreifende Benutzer eine neue Verbindung zum Cluster herstellen.

1. Öffnen Sie die Amazon-MSK-Konsole unter AWS Management Console.
2. Wählen Sie im Navigationsbereich unter MSK-Cluster die Option Cluster aus, um die Liste der Cluster im Konto anzuzeigen.
3. Wählen Sie die Cluster aus, die Sie bearbeiten möchten. Der Cluster muss sich im ACTIVE-Status befinden.
4. Wählen Sie die Registerkarte Eigenschaften des Clusters und wechseln Sie dann zu Netzwerkeinstellungen.

5. Wählen Sie das Dropdown-Menü Bearbeiten und dann Multi-VPC-Konnektivität deaktivieren (um ein Authentifizierungsschema auszuschalten).
6. Wählen Sie einen oder mehrere Authentifizierungstyp(en) aus, die Sie für diesen Cluster deaktivieren möchten.
7. Wählen Sie Auswahl deaktivieren.

Example So schalten Sie ein Authentifizierungsschema mit der API ein-/aus

Als Alternative zur MSK-Konsole können Sie die UpdateConnectivity API verwenden, um private Multi-VPC-Konnektivität zu aktivieren und Authentifizierungsschemata auf einem ACTIVE-Cluster zu konfigurieren. Das folgende Beispiel zeigt, dass SASL/SCRAM- und IAM-Authentifizierungsschema für den Cluster aktiviert sind.

Wenn Sie ein neues Authentifizierungsschema aktivieren, sollten Sie auch neue verwaltete VPC-Verbindungen für das neue Authentifizierungsschema erstellen und Ihre Clients so aktualisieren, dass sie die für das neue Authentifizierungsschema spezifischen Bootstrap-Broker verwenden.

Wenn Sie private Multi-VPC-Konnektivität für Authentifizierungsschemas deaktivieren, ändern sich bestehende VPC-Verbindungen auf der Client-Seite in INACTIVE, und die PrivateLink-Infrastruktur auf der Cluster-Seite, einschließlich der verwalteten VPC-Verbindungen, wird entfernt. Der kontoübergreifende Benutzer kann nur die inaktive VPC-Verbindung löschen. Wenn die private Konnektivität auf dem Cluster wieder aktiviert wird, muss der kontoübergreifende Benutzer eine neue Verbindung zum Cluster herstellen.

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

```
          }
        },
        "tls": {
          "enabled": FALSE
        }
      }
    }
  }
}

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

Eine verwaltete VPC-Verbindung zu einem Amazon-MSK-Cluster ablehnen

Von der Amazon-MSK-Konsole auf dem Cluster-Administratorkonto aus können Sie eine Client-VPC-Verbindung ablehnen. Die Client-VPC-Verbindung muss sich im Status AVAILABLE befinden, damit sie abgelehnt werden kann. Möglicherweise möchten Sie eine verwaltete VPC-Verbindung von einem Client ablehnen, der nicht mehr autorisiert ist, eine Verbindung zu Ihrem Cluster herzustellen. Um zu verhindern, dass neue verwaltete VPC-Verbindungen eine Verbindung zu einem Client herstellen, verweigern Sie den Zugriff auf den Client in der Cluster-Richtlinie. Eine abgelehnte Verbindung verursacht immer noch Kosten, bis sie vom Verbindungsbesitzer gelöscht wird. Siehe Löschen einer verwalteten VPC-Verbindung zu einem Amazon-MSK-Cluster.

So lehnen Sie eine Client-VPC-Verbindung mithilfe der MSK-Konsole ab

1. Öffnen Sie die Amazon-MSK-Konsole unter AWS Management Console.
2. Wählen Sie im Navigationsbereich Cluster aus und scrollen Sie zu der Liste Netzwerkeinstellungen > Client-VPC-Verbindungen.
3. Wählen Sie die Verbindung aus, die Sie ablehnen möchten, und wählen Sie Client-VPC-Verbindung ablehnen.
4. Bestätigen Sie, dass Sie die ausgewählte Client-VPC-Verbindung ablehnen möchten.

Verwenden Sie die `RejectClientVpcConnection`-API, um eine verwaltete VPC-Verbindung mithilfe der API abzulehnen.

Eine verwaltete VPC-Verbindung zu einem Amazon-MSK-Cluster löschen

Der kontoübergreifende Benutzer kann eine verwaltete VPC-Verbindung für einen MSK-Cluster von der Konsole des Client-Kontos aus löschen. Da der Benutzer des Cluster-Besitzers nicht Eigentümer der verwalteten VPC-Verbindung ist, kann die Verbindung nicht aus dem Cluster-Administratorkonto gelöscht werden. Sobald eine VPC-Verbindung gelöscht wurde, fallen keine Kosten mehr an.

So löschen Sie eine verwaltete VPC-Verbindung mit der MSK-Konsole

1. Öffnen Sie vom Client-Konto aus die Amazon-MSK-Konsole unter AWS Management Console.
2. Wählen Sie im Navigationsbereich Verwaltete VPC-Verbindungen.
3. Wählen Sie in der Liste der Verbindungen die Verbindung aus, die Sie löschen möchten.
4. Bestätigen Sie, dass Sie die VPC-Verbindung löschen möchten.

Verwenden Sie die `DeleteVpcConnection`-API, um eine verwaltete VPC-Verbindung mithilfe der API zu löschen.

Berechtigungen für private Multi-VPC-Konnektivität

In diesem Abschnitt werden die Berechtigungen zusammengefasst, die für Clients und Cluster erforderlich sind, die die private Multi-VPC-Konnektivitäts-Feature verwenden. Private Multi-VPC-Konnektivität erfordert, dass der Client-Administrator für jeden Client, der über eine verwaltete VPC-Verbindung zum MSK-Cluster verfügt, Berechtigungen erstellt. Außerdem muss der MSK-Clusteradministrator die PrivateLink Konnektivität auf dem MSK-Cluster aktivieren und Authentifizierungsschemata auswählen, um den Zugriff auf den Cluster zu kontrollieren.

Cluster-Authentifizierungstyp und Zugriffsberechtigungen für Themen

Aktivieren Sie die private Multi-VPC-Konnektivitäts-Feature für Authentifizierungsschemas, die für Ihren MSK-Cluster aktiviert sind. Siehe Anforderungen und Einschränkungen für private Multi-VPC-Konnektivität. Wenn Sie Ihren MSK-Cluster für die Verwendung des SASL/SCRAM-Authentifizierungsschemas konfigurieren, ist die Apache-Kafka-ACLs-Eigenschaft `allow.everyone.if.no.acl.found=false` obligatorisch. Nachdem Sie die Apache Kafka ACLs für Ihren Cluster festgelegt haben, aktualisieren Sie die Cluster-Konfiguration, sodass die Eigenschaft `allow.everyone.if.no.acl.found` für den Cluster auf Falsch gesetzt wird. Weitere Informationen zum Aktualisieren der Konfiguration eines Clusters finden Sie unter Amazon-MSK-Konfigurationsvorgänge.

Kontoübergreifende Cluster-Richtlinienberechtigungen

Wenn sich ein Kafka-Client in einem anderen AWS Konto als dem MSK-Cluster befindet, fügen Sie dem MSK-Cluster eine clusterbasierte Richtlinie hinzu, die den Client-Root-Benutzer für kontoübergreifende Konnektivität autorisiert. Sie können die Multi-VPC-Cluster-Richtlinie mit dem IAM-Richtlinien-Editor in der MSK-Konsole bearbeiten (Cluster Sicherheitseinstellungen > Cluster-Richtlinie bearbeiten) oder die folgenden APIs verwenden, um die Cluster-Richtlinie zu verwalten:

PutClusterRichtlinie

Hängt eine Cluster-Richtlinie an den MSK-Cluster an. Sie können diese API verwenden, um die angegebene MSK-Cluster-Richtlinie zu erstellen oder zu aktualisieren. Wenn Sie die Richtlinie aktualisieren, ist das Feld currentVersion in der Nutzlast der Anfrage erforderlich.

GetClusterPolitik

Ruft den JSON-Text des Cluster-Richtliniendokuments ab, das an den Cluster angehängt ist.

DeleteClusterPolitik

Löscht die Cluster-Richtlinie.

Als Referenz finden Sie im Folgenden ein JSON-Beispiel für eine grundlegende Cluster-Richtlinie, ähnlich der, die im IAM-Richtlinien-Editor der MSK-Konsole angezeigt wird.

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

```
        }
    ]
}
```

Client-Berechtigungen für private Multi-VPC-Konnektivität zu einem MSK-Cluster

Um private Multi-VPC-Konnektivität zwischen einem Kafka-Client und einem MSK-Cluster einzurichten, benötigt der Client eine angehängte Identitätsrichtlinie, die Berechtigungen für die Aktionen `kafka:CreateVpcConnection`, `ec2:CreateTags` und `ec2:CreateVPCEndpoint` für den Client gewährt. Zum Nachschlagen finden Sie nachstehend ein JSON-Beispiel für eine grundlegende Client-Identitätsrichtlinie.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kafka:CreateVpcConnection",
        "ec2:CreateTags",
        "ec2:CreateVPCEndpoint"
      ],
      "Resource": "*"
    }
  ]
}
```

Port-Informationen

Verwenden Sie die folgenden Portnummern, damit Amazon MSK mit Client-Computern kommunizieren kann:

- Um mit Brokern in Klartext zu kommunizieren, verwenden Sie Port 9092.
- Um mit Brokern mit TLS-Verschlüsselung zu kommunizieren, verwenden Sie Port 9094 für den Zugriff von innen AWS und Port 9194 für den öffentlichen Zugriff.
- Um mit Brokern über SASL/SCRAM zu kommunizieren, verwenden Sie Port 9096 für den Zugriff von innen AWS und Port 9196 für den öffentlichen Zugriff.
- Um mit Brokern in einem Cluster zu kommunizieren, der für die Nutzung eingerichtet istthe section called “IAM-Zugriffssteuerung”, verwenden Sie Port 9098 für den Zugriff von innen AWS und Port 9198 für den öffentlichen Zugriff.

- Verwenden Sie Port 2182, um mit Apache ZooKeeper mithilfe der TLS-Verschlüsselung zu kommunizieren. ZooKeeper Apache-Knoten verwenden standardmäßig Port 2181.

Migration zu einem Amazon-MSK-Cluster

Amazon MSK Replicator kann für die MSK-Cluster-Migration verwendet werden. Siehe Was ist Amazon MSK Replicator?. Alternativ können Sie Apache MirrorMaker 2.0 verwenden, um von einem Nicht-MSK-Cluster zu einem Amazon MSK-Cluster zu migrieren. Ein Beispiel dafür finden Sie unter Migrieren eines lokalen Apache Kafka-Clusters zu Amazon MSK mithilfe von. MirrorMaker Informationen zur Verwendung MirrorMaker finden Sie unter Spiegeln von Daten zwischen Clustern in der Apache Kafka-Dokumentation. Wir empfehlen die Einrichtung MirrorMaker in einer Konfiguration mit hoher Verfügbarkeit.

Eine Übersicht der Schritte, die bei der Migration MirrorMaker zu einem MSK-Cluster zu befolgen sind

1. Erstellen Sie den MSK-Ziel-Cluster
2. Starten Sie mit MirrorMaker einer Amazon EC2 EC2-Instance innerhalb derselben Amazon VPC wie der Ziel-Cluster.
3. Untersuchen Sie die Verzögerung MirrorMaker .
4. Leiten MirrorMaker Sie nach dem Aufholen die Produzenten und Verbraucher mithilfe der MSK-Cluster-Bootstrap-Broker zum neuen Cluster um.
5. Herunterfahren. MirrorMaker

Migrieren Ihres Apache-Kafka-Clusters zu Amazon MSK

Angenommen, Sie haben einen Apache-Kafka-Cluster namens `CLUSTER_ONPREM`. Dieser Cluster wird mit Themen und Daten gefüllt. Wenn Sie diesen Cluster zu einem neu erstellten Amazon-MSK-Cluster mit dem Namen `CLUSTER_AWSMSK` migrieren möchten, bietet dieses Verfahren eine allgemeine Ansicht der auszuführenden Schritte.

So migrieren Sie Ihren vorhandenen Apache-Kafka-Cluster zu Amazon MSK

1. Erstellen Sie in `CLUSTER_AWSMSK` alle Themen, die Sie migrieren möchten.

   Sie können diesen Schritt nicht verwenden MirrorMaker , da er die Themen, die Sie migrieren möchten, nicht automatisch mit der richtigen Replikationsebene neu erstellt. Sie können die Themen in Amazon MSK mit denselben Replikationsfaktoren und der Anzahl von Partitionen wie in `CLUSTER_ONPREM` erstellen. Sie können die Themen auch mit unterschiedlichen Replikationsfaktoren und Partitionszahlen erstellen.

2. Beginnen Sie mit MirrorMaker einer Instanz, die Lese CLUSTER_ONPREM - und Schreibzugriff CLUSTER_AWSMSK hat.
3. Führen Sie den folgenden Befehl aus, um alle Themen zu spiegeln:

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

   In diesem Befehl weist `config/mirrormaker-consumer.properties` auf einen Bootstrap-Broker in CLUSTER_ONPREM (z. B. `bootstrap.servers=localhost:9092`). Und `config/mirrormaker-producer.properties` zeigt auf einen Bootstrap-Broker in CLUSTER_AWSMSK; zum Beispiel. `bootstrap.servers=10.0.0.237:9092,10.0.2.196:9092,10.0.1.233:9092`
4. Lassen Sie es im Hintergrund MirrorMaker laufen und verwenden Sie es weiter. CLUSTER_ONPREM MirrorMaker spiegelt alle neuen Daten wider.
5. Überprüfen Sie den Fortschritt der Spiegelung, indem Sie die Verzögerung zwischen dem letzten Offset für jedes Thema und dem aktuellen Offset überprüfen, ab dem die Spiegelung verbraucht MirrorMaker wird.

   Denken Sie daran, MirrorMaker dass Sie lediglich einen Verbraucher und einen Hersteller verwenden. So können Sie die Verzögerung mit dem `kafka-consumer-groups.sh`-Werkzeug überprüfen. Um den Namen der Verbrauchergruppe zu finden, suchen Sie in der `mirrormaker-consumer.properties`-Datei nach der `group.id` und verwenden Sie den Wert. Wenn es keinen solchen Schlüssel in der Datei gibt, können Sie ihn erstellen. Legen Sie beispielsweise `group.id=mirrormaker-consumer-group` fest.
6. Wenn Sie mit dem Spiegeln aller Themen MirrorMaker fertig sind, beenden Sie alle Produzenten und Verbraucher und hören Sie dann auf MirrorMaker. Leiten Sie dann die Produzenten und Konsumenten in den CLUSTER_AWSMSK-Cluster um, indem Sie die Werte der Produzenten und Konsumenten des Bootstrap-Brokers ändern. Starten Sie alle Produzenten und Konsumenten auf CLUSTER_AWSMSK neu.

Migration zwischen zwei Amazon-MSK-Clustern

Sie können Apache MirrorMaker 2.0 verwenden, um von einem Nicht-MSK-Cluster zu einem MSK-Cluster zu migrieren. Sie können beispielsweise von einer Version von Apache Kafka zu einer anderen migrieren. Ein Beispiel dafür finden Sie unter Migrieren eines lokalen Apache Kafka-Clusters

zu Amazon MSK mithilfe von. MirrorMaker Als Alternative kann Amazon MSK Replicator für die MSK-Cluster-Migration verwendet werden. Weitere Informationen über Amazon MSK Replicator finden Sie unter MSK-Replikator.

MirrorMaker 1.0 bewährte Methoden

Diese Liste mit bewährten Methoden gilt für MirrorMaker 1.0.

- MirrorMaker Auf dem Zielcluster ausführen. Wenn ein Netzwerkproblem auftritt, sind die Nachrichten auf diese Weise weiterhin im Quell-Cluster verfügbar. Wenn Sie MirrorMaker auf dem Quellcluster ausführen und Ereignisse im Producer zwischengespeichert werden und es ein Netzwerkproblem gibt, gehen Ereignisse möglicherweise verloren.
- Wenn während der Übertragung eine Verschlüsselung erforderlich ist, führen Sie diese im Quell-Cluster aus.
- Legen Sie für Konsumenten „auto.commit.enabled=false“ fest.
- Für Produzenten legen Sie Folgendes fest:
  - max.in.flight.requests.per.connection=1
  - retries=Int.Max_Value
  - acks=all
  - max.block.ms = Long.Max_Value
- Für einen hohen Produzentendurchsatz:
  - Nachrichten puffern und Nachrichten-Batches füllen – buffer.memory, batch.size, linger.ms optimieren
  - Socket-Puffer optimieren – receive.buffer.bytes, send.buffer.bytes
- Um Datenverlust zu vermeiden, schalten Sie das auto Commit an der Quelle aus, sodass die Commits gesteuert werden MirrorMaker können. Dies geschieht normalerweise, nachdem es das ACK vom Zielcluster erhalten hat. Wenn der Producer acks=all und der Zielcluster min.insync.replicas auf mehr als 1 gesetzt hat, werden die Nachrichten auf mehr als einem Broker am Ziel gespeichert, bevor der Verbraucher den Offset an der Quelle festschreibt. MirrorMaker
- Wenn die Reihenfolge wichtig ist, können Sie die Wiederholungsversuche auf „0“ festlegen. Setzen Sie die maximalen Inflight-Verbindungen für eine Produktionsumgebung alternativ auf „1“, um sicherzustellen, dass die Commits für die versendeten Stapel in der richtigen Reihenfolge durchgeführt werden, falls ein Stapel in der Mitte ausfällt. Auf diese Weise wird jeder gesendete Stapel wiederholt, bis der nächste Stapel gesendet wird. Wenn „max.block.ms“ nicht auf den Maximalwert festgelegt ist und der Puffer des Produzenten voll ist, kann es zu Datenverlust

kommen (abhängig von einigen der anderen Einstellungen). Dies kann den Konsumenten blockieren und Druck erzeugen.

- Für hohen Durchsatz
  - Erhöhen Sie den Pufferspeicher.
  - Erhöhen Sie die Stapelgröße.
  - Passen Sie linger.ms an, damit die Stapel gefüllt werden können. Dies ermöglicht zudem eine bessere Komprimierung, weniger Auslastung der Netzwerkbandbreite und weniger Speicher auf dem Cluster. Dies führt zu einer erhöhten Retention.
  - Überwachen Sie die CPU- und Speichernutzung.
- Für hohen Konsumentendurchsatz
  - Erhöhen Sie MirrorMaker die Anzahl der Threads/Verbraucher pro Prozess — num.streams.
  - Erhöhen Sie zunächst die Anzahl der MirrorMaker Prozesse auf allen Computern, bevor Sie die Anzahl der Threads erhöhen, um eine hohe Verfügbarkeit zu gewährleisten.
  - Erhöhen Sie die Anzahl der MirrorMaker Prozesse zuerst auf demselben Computer und dann auf verschiedenen Computern (mit derselben Gruppen-ID).
  - Isolieren Sie Themen mit sehr hohem Durchsatz und verwenden Sie separate MirrorMaker Instanzen.
- Für Verwaltung und Konfiguration
  - AWS CloudFormation Verwendungs- und Konfigurationsmanagement-Tools wie Chef und Ansible.
  - Verwenden Sie Amazon-EFS-Mounts, um alle Konfigurationsdateien von allen Amazon-EC2-Instances aus zugänglich zu machen.
  - Verwenden Sie Container für die einfache Skalierung und Verwaltung von MirrorMaker Instanzen.
- In der Regel braucht es mehr als einen Verbraucher, um einen Hersteller zu überzeugen. MirrorMaker Richten Sie also mehrere Konsumenten ein. Richten Sie sie zunächst auf verschiedenen Computern ein, um eine hohe Verfügbarkeit zu gewährleisten. Skalieren Sie dann einzelne Computer bis zu einem Konsumenten pro Partition, wobei die Konsumenten gleichmäßig auf die Computer verteilt sind.
- Da die Standardwerte möglicherweise zu niedrig sind, optimieren Sie die Puffer für Empfangen und Senden, um einen hohen Durchsatz zu erreichen. Um eine maximale Leistung zu erzielen, stellen Sie sicher, dass die Gesamtzahl der Streams (num.streams) mit allen Themenpartitionen übereinstimmt, MirrorMaker die versucht werden, in den Zielcluster zu kopieren.

MirrorMaker 2.* Vorteile

- Nutzt das Apache Kafka Connect-Framework und -Partnersystem.
- Erkennt neue Themen und Partitionen.
- Synchronisiert die Themenkonfiguration automatisch zwischen Clustern.
- Unterstützt „aktiv/aktiv“-Clusterpaare sowie eine beliebige Anzahl aktiver Cluster.
- Bietet neue Metriken, einschließlich der end-to-end Replikationslatenz über mehrere Rechenzentren und Cluster hinweg.
- Gibt Offsets aus, die für die Migration von Konsumenten zwischen Clustern erforderlich sind, und stellt Werkzeuge für die Offset-Übertragung bereit.
- Unterstützt eine Konfigurationsdatei auf hoher Ebene, mit der mehrere Cluster und Replikationsabläufe an einem zentralen Ort spezifiziert werden können, im Vergleich zu den Eigenschaften auf niedriger Ebene für jeden MirrorMaker 1.*-Prozess.

Überwachung eines Amazon-MSK-Clusters

Es gibt mehrere Möglichkeiten, wie Amazon MSK Ihnen bei der Überwachung des Status Ihres Amazon-MSK-Clusters hilft.

- Amazon MSK unterstützt Sie bei der Überwachung Ihrer Festplattenspeicherkapazität, indem es Ihnen automatisch Speicherkapazitätswarnungen sendet, wenn ein Cluster kurz davor ist, seine Speicherkapazitätsgrenze zu erreichen. Die Warnmeldungen enthalten auch Empfehlungen zu den besten Maßnahmen zur Behebung festgestellter Probleme. Auf diese Weise können Sie Festplattenkapazitätsprobleme erkennen und schnell beheben, bevor sie kritisch werden. Amazon MSK sendet diese Benachrichtigungen automatisch an die Amazon MSK-Konsole AWS Health Dashboard EventBridge, Amazon und E-Mail-Kontakte für Ihr AWS Konto. Weitere Informationen zu Warnmeldungen zur Speicherkapazität finden Sie unter Amazon-MSK-Speicherkapazitätswarnungen.
- Amazon MSK sammelt Apache Kafka-Metriken und sendet sie an Amazon, CloudWatch wo Sie sie einsehen können. Weitere Informationen zu Apache-Kafka-Metriken, einschließlich derjenigen, die von Amazon MSK angezeigt werden, finden Sie unter Überwachung in der Apache-Kafka-Dokumentation.
- Sie können Ihren MSK-Cluster auch mit Prometheus, einer Open-Source-Überwachungsanwendung, überwachen. Weitere Informationen zu Prometheus finden Sie unter Overview in der Prometheus-Dokumentation. Informationen zur Überwachung Ihres Clusters mit Prometheus finden Sie unter the section called “Offene Überwachung mit Prometheus”.

Themen

Amazon MSK-Metriken für die Überwachung mit CloudWatch

Amazon MSK ist in Amazon integriert, CloudWatch sodass Sie CloudWatch Metriken für Ihren Amazon MSK-Cluster sammeln, anzeigen und analysieren können. Die Metriken, die Sie für Ihren

MSK-Cluster konfigurieren, werden automatisch gesammelt und weitergeleitet. CloudWatch Sie können die Überwachungsebene für einen MSK-Cluster auf eine der folgenden Stufen festlegen: `DEFAULT`, `PER_BROKER`, `PER_TOPIC_PER_BROKER` oder `PER_TOPIC_PER_PARTITION`. Die Tabellen im folgenden Abschnitt zeigen alle Metriken, die in jeder Überwachungsebene verfügbar sind.

> **Note**
>
> Die Namen einiger Amazon MSK-Metriken für die CloudWatch Überwachung haben sich in Version 3.6.0 und höher geändert. Verwenden Sie die neuen Namen für die Überwachung dieser Metriken. Für Metriken mit geänderten Namen zeigt die nachfolgende Tabelle den Namen, der in Version 3.6.0 und höher verwendet wurde, gefolgt vom Namen in Version 2.8.2.tiered.

Metriken auf der `DEFAULT`-Ebene sind kostenlos. Die Preise für andere Kennzahlen sind auf der CloudWatchAmazon-Preisseite beschrieben.

Überwachung auf **`DEFAULT`**-Ebene

Die in der folgenden Tabelle beschriebenen Metriken sind auf der `DEFAULT`-Überwachungsebene verfügbar. Sie sind kostenlos.

Auf der **`DEFAULT`**-Überwachungsebene verfügbare Metriken

Name	Wenn sichtbar	Dimensi en	Beschreibung
`ActiveCon trollerCount`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name	Zu jeder Zeit sollte nur ein Controller pro Cluster aktiv sein.
`BurstBalance`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Der verbleibende Saldo der Eingabe-Ausgabe-Burst-Credits für EBS-Volumes im Cluster. Verwenden Sie dies, um Latenz oder verringerten Durchsatz zu untersuchen.

Name	Wenn sichtbar	Dimensionen	Beschreibung
			`BurstBalance` wird für EBS-Volumes nicht berichtet, wenn die Basisleistung eines Volumes höher als die maximale Burst-Leistung ist. Weitere Informationen zur Funktionsweise von Burst-Gutschriften in finden Sie unter I/O-Guthaben und Burst-Performance.
`BytesInPerSec`	Nachdem Sie ein Thema erstellt haben.	Cluster-Name, Broker-ID, Thema	Die Anzahl der Bytes, die pro Sekunde von Clients empfangen werden. Diese Metrik ist pro Broker und auch pro Thema verfügbar.
`BytesOutPerSec`	Nachdem Sie ein Thema erstellt haben.	Cluster-Name, Broker-ID, Thema	Die Anzahl der Bytes, die pro Sekunde an Clients gesendet werden. Diese Metrik ist pro Broker und auch pro Thema verfügbar.
`ClientConnectionCount`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID, Client-Authentifizierung	Die Anzahl der aktiven authentifizierten Client-Verbindungen.
`ConnectionCount`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Anzahl der aktiven authentifizierten und nicht authentifizierten Verbindungen sowie Verbindungen zwischen Brokern.

Name	Wenn sichtbar	Dimensionen	Beschreibung
`CPUCreditBalance`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Anzahl verdienter CPU-Guthaben, die ein Broker angesammelt hat, seit er gestartet wurde. Guthaben werden auf dem Guthaben-Konto angesammelt, nachdem sie verdient wurden, und davon entfernt, wenn sie verbraucht werden. Wenn das CPU-Guthaben aufgebraucht ist, kann sich dies negativ auf die Leistung Ihres Clusters auswirken. Sie können Maßnahmen ergreifen, um die CPU-Last zu reduzieren. Sie können beispielsweise die Anzahl der Client-Anfragen reduzieren oder den Broker-Typ auf einen M5-Broker-Typ aktualisieren.
`CpuIdle`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Der Anteil der CPU-Leerlaufzeit.
`CpuIoWait`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Der Prozentsatz der CPU-Leerlaufzeit während eines ausstehenden Festplattenvorgangs.
`CpuSystem`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Der Anteil der CPU im Kernel-Speicher.

Name	Wenn sichtbar	Dimensionen	Beschreibung
`CpuUser`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Der Anteil der CPU im Benutzerbereich.
`GlobalPartitionCount`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name	Die Anzahl der Partitionen für alle Themen im Cluster, ausgenommen Replikate. Da `GlobalPartitionCount` keine Replikate enthalten sind, kann die Summe der `PartitionCount` Werte höher sein, als GlobalPartitionCount wenn der Replikationsfaktor für ein Thema größer als 1 ist.
`GlobalTopicCount`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name	Gesamtzahl der Themen für alle Broker im Cluster.
`EstimatedMaxTimeLag`	Nachdem die Verbrauchergruppe von einem Thema konsumiert hat.	Verbrauchergruppe Thema	Voraussichtlicher Zeitaufwand (in Sekunden) bis zur Entleerung von `MaxOffsetLag` .
`KafkaAppLogsDiskUsed`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Der Anteil des Festplattenspeichers, der für Anwendungsprotokolle verwendet wird.
`KafkaDataLogsDiskUsed (Cluster Name, Broker ID-Dimension)`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Der Anteil des Festplattenspeichers, der für Datenprotokolle verwendet wird.

Name	Wenn sichtbar	Dimensionen	Beschreibung
`KafkaDataLogsDiskUsed` (`Cluster Name`-Dimension)	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name	Der Anteil des Festplattenspeichers, der für Datenprotokolle verwendet wird.
`LeaderCount`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Gesamtzahl der Partitionsleiter pro Broker, ohne Replikate.
`MaxOffsetLag`	Nachdem die Verbrauchergruppe von einem Thema konsumiert hat.	Verbrauchergruppe Thema	Die maximale Offset-Verzögerung für alle Partitionen in einem Thema.
`MemoryBuffered`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Größe des gepufferten Arbeitsspeichers in Bytes für den Broker.
`MemoryCached`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Größe des zwischengespeicherten Arbeitsspeichers in Bytes für den Broker.
`MemoryFree`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Arbeitsspeichergröße in Byte, die frei und für den Broker verfügbar ist.
`HeapMemoryAfterGC`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Der Prozentsatz des gesamten Heap-Speichers, der nach der Garbage Collection verwendet wird.

Name	Wenn sichtbar	Dimensi en	Beschreibung
`MemoryUsed`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Größe des Arbeitsspeichers in Byte, der für den Broker verwendet wird.
`MessagesInPerSec`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Anzahl der Nachrichten, die pro Sekunde für den Broker eingehen.
`NetworkRxDropped`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Anzahl der gelöschten Empfangspakete.
`NetworkRxErrors`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Anzahl der Netzwerkempfangsfehler für den Broker.
`NetworkRxPackets`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Anzahl der vom Broker empfangenen Pakete.
`NetworkTxDropped`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Anzahl der gelöschten Übertragungspakete.
`NetworkTxErrors`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Anzahl der Netzwerkübertragungsfehler für den Broker.

Name	Wenn sichtbar	Dimensionen	Beschreibung
`NetworkTxPackets`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Anzahl der vom Broker übertragenen Pakete.
`OfflinePartitionsCount`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name	Die Gesamtzahl der Partitionen, die im Cluster offline sind.
`PartitionCount`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Gesamtzahl der Themenpartitionen pro Broker, einschließlich Replikate.
`ProduceTotalTimeMsMean`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die mittlere Erzeugungszeit in Millisekunden.
`RequestBytesMean`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die mittlere Anzahl der Anforderungs-Bytes für den Broker.
`RequestTime`	Nachdem die Anforderungsablehnung angewendet wurde.	Cluster-Name, Broker-ID	Die durchschnittliche Zeit (in Millisekunden) für die Verarbeitung von Anforderungen in Broker-Netzwerk- und E/A-Threads.
`RootDiskUsed`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Der Anteil der vom Broker verwendeten Stamm-Datenträger.

Name	Wenn sichtbar	Dimensi en	Beschreibung
SumOffsetLag	Nachdem die Verbrauchergruppe von einem Thema konsumiert hat.	Verbrauc ergruppe Thema	Die aggregierte Offset-Verzögerung für alle Partitionen in einem Thema.
SwapFree	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Größe des für den Broker verfügbaren Auslagerungsspeichers in Bytes.
SwapUsed	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Größe des Auslagerungsspeich ers in Bytes, der für den Broker verwendet wird.
TrafficShaping	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Allgemeine Metriken, die die Anzahl der Pakete angeben, die aufgrund von Überschreitungen der Netzwerkz uweisungen geformt (verworfen oder in die Warteschlange gestellt) wurden. Genauere Details sind mit PER_BROKER-Metriken verfügbar.
UnderMinI srPartiti onCount	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Anzahl der „under minIsr“-P artitionen für den Broker.
UnderRepl icatedPar titions	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Die Anzahl der nicht replizierten Partitionen für den Broker.

Name	Wenn sichtbar	Dimensi en	Beschreibung
`ZooKeeper RequestLa tencyMsMean`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Für einen ZooKeeper basierten Cluster. Die durchschnittliche Latenz in Millisekunden für ZooKeeper Apache-Anfragen vom Broker.
`ZooKeeper SessionState`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Cluster-Name, Broker-ID	Für ZooKeeper einen basierten Cluster. Verbindungsstatus der ZooKeeper Brokersitzung, der einer der folgenden sein kann: NOT_CONNECTED: '0.0', ASSOCIATING: '0.1', CONNECTING: '0.5', CONNECTEDREADONLY: '0.8', CONNECTED: '1.0', CLOSED: '5.0', AUTH_FAILED: '10.0'.

Überwachung auf **`PER_BROKER`**-Ebene

Wenn Sie die Überwachungsebene auf „`PER_BROKER`“ festlegen, erhalten Sie die in der folgenden Tabelle beschriebenen Metriken zusätzlich zu allen `DEFAULT`-Ebenenmetriken. Sie zahlen für die Metriken in der folgenden Tabelle. Die `DEFAULT`-Ebenenmetriken sind allerdings weiterhin kostenlos. Die Metriken in dieser Tabelle haben die folgenden Dimensionen: Clustername, Broker-ID.

Zusätzliche Metriken, die ab der **`PER_BROKER`**-Überwachungsebene verfügbar sind

Name	Wenn sichtbar	Beschreibung
`BwInAllowanceExceeded`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Anzahl der Pakete, die geformt wurden, weil die eingehende aggregier te Bandbreite das Maximum für den Broker überschritten hat.
`BwOutAllowanceExce eded`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Anzahl der Pakete, die geformt wurden, weil die ausgehende

Name	Wenn sichtbar	Beschreibung
		aggregierte Bandbreite das Maximum für den Broker überschritten hat.
`ConnTrackAllowanceExceeded`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Anzahl der Pakete, die geformt wurden, weil die Verbindungs-Nachverfolgung das Maximum für den Broker überschritten hat. Die Verbindungs-Nachverfolgung is mit Sicherheitsgruppen verbunden, die jede aufgebaute Verbindung nachverfolgen, um sicherzustellen, dass Retour-Pakete wie erwartet bereitgestellt werden.
`ConnectionCloseRate`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Anzahl der pro Sekunde und Listener geschlossenen Verbindungen. Diese Zahl wird pro Listener aggregiert und nach den Client-Listenern gefiltert.
`ConnectionCreationRate`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Anzahl der neuen Verbindungen, die pro Sekunde und Listener hergestellt werden. Diese Zahl wird pro Listener aggregiert und nach den Client-Listenern gefiltert.
`CpuCreditUsage`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Das vom Broker verbrauchte CPU-Guthaben. Wenn das CPU-Guthaben aufgebraucht ist, kann sich dies negativ auf die Leistung Ihres Clusters auswirken. Sie können Maßnahmen ergreifen, um die CPU-Last zu reduzieren. Sie können beispielsweise die Anzahl der Client-Anfragen reduzieren oder den Broker-Typ auf einen M5-Broker-Typ aktualisieren.

Name	Wenn sichtbar	Beschreibung
FetchConsumerLocalTimeMsMean	Nachdem ein Produzent/Konsument vorhanden ist.	Die mittlere Zeit in Millisekunden, die die Konsumentenanforderung beim Leader verarbeitet wird.
FetchConsumerRequestQueueTimeMsMean	Nachdem ein Produzent/Konsument vorhanden ist.	Die mittlere Zeit in Millisekunden, die sich die Konsumentenanforderung in der Anforderungswarteschlange befindet.
FetchConsumerResponseQueueTimeMsMean	Nachdem ein Produzent/Konsument vorhanden ist.	Die mittlere Zeit in Millisekunden, die sich die Konsumentenanforderung in der Antwortwarteschlange befindet.
FetchConsumerResponseSendTimeMsMean	Nachdem ein Produzent/Konsument vorhanden ist.	Die mittlere Zeit in Millisekunden in der der Verbraucher eine Antwort sendet.
FetchConsumerTotalTimeMsMean	Nachdem ein Produzent/Konsument vorhanden ist.	Die mittlere Gesamtzeit in Millisekunden, die Konsumenten für das Abrufen von Daten vom Broker benötigen.
FetchFollowerLocalTimeMsMean	Nachdem ein Produzent/Konsument vorhanden ist.	Die mittlere Zeit in Millisekunden, in der die Follower-Anforderung beim Leader verarbeitet wird.
FetchFollowerRequestQueueTimeMsMean	Nachdem ein Produzent/Konsument vorhanden ist.	Die mittlere Zeit in Millisekunden, die sich die Follower-Anforderung in der Anforderungswarteschlange befindet.
FetchFollowerResponseQueueTimeMsMean	Nachdem ein Produzent/Konsument vorhanden ist.	Die mittlere Zeit in Millisekunden, die sich die Follower-Anforderung in der Antwortwarteschlange befindet.
FetchFollowerResponseSendTimeMsMean	Nachdem ein Produzent/Konsument vorhanden ist.	Die mittlere Zeit in Millisekunden, in der der Follower eine Antwort sendet.

Name	Wenn sichtbar	Beschreibung
FetchFollowerTotalTimeMsMean	Nachdem ein Produzent/Konsument vorhanden ist.	Die mittlere Gesamtzeit in Millisekunden, die Follower für das Abrufen von Daten vom Broker benötigen.
FetchMessageConversionsPerSec	Nachdem Sie ein Thema erstellt haben.	Die Anzahl der Abrufnachrichtenkonvertierungen pro Sekunde für den Broker.
FetchThrottleByteRate	Nachdem die Bandbreitenablehnung angewendet wurde.	Die Anzahl der gedrosselten Bytes pro Sekunde.
FetchThrottleQueueSize	Nachdem die Bandbreitenablehnung angewendet wurde.	Die Anzahl der Nachrichten in der Drosselungswarteschlange.
FetchThrottleTime	Nachdem die Bandbreitenablehnung angewendet wurde.	Die durchschnittliche Abrufdrosselzeit in Millisekunden.
IAMNumberOfConnectionRequests	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Anzahl der IAM-Authentifizierungsanfragen pro Sekunde.
IAMTooManyConnections	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Anzahl der versuchten Verbindungen liegt über 100. 0 bedeutet, dass die Anzahl der Verbindungen innerhalb des Grenzwerts liegt. Wenn >0, wird die Drosselungsgrenze überschritten und Sie müssen die Anzahl der Verbindungen reduzieren.

Name	Wenn sichtbar	Beschreibung
NetworkProcessorAvgIdlePercent	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Der durchschnittliche Anteil der Zeit, die sich die Netzwerkprozessoren im Leerlauf befinden.
PpsAllowanceExceeded	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Anzahl der Pakete, die geformt wurden, weil die bidirektionale PPS das Maximum für den Broker überschritten hat.
ProduceLocalTimeMsMean	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die durchschnittliche Zeit in Millisekunden, in der die Anfrage beim Leader verarbeitet wird.
ProduceMessageConversionsPerSec	Nachdem Sie ein Thema erstellt haben.	Die Anzahl der Erzeugnisnachrichtenkonvertierungen pro Sekunde für den Broker.
ProduceMessageConversionsTimeMsMean	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die mittlere Zeit in Millisekunden für Nachrichtenformatkonvertierungen.
ProduceRequestQueueTimeMsMean	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die mittlere Zeit in Millisekunden, die sich Anforderungsnachrichten in der Warteschlange befinden.
ProduceResponseQueueTimeMsMean	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die mittlere Zeit in Millisekunden, die sich Antwortnachrichten in der Warteschlange befinden.
ProduceResponseSendTimeMsMean	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die mittlere Zeit in Millisekunden für das Senden von Antwortnachrichten.
ProduceThrottleByteRate	Nachdem die Bandbreitenablehnung angewendet wurde.	Die Anzahl der gedrosselten Bytes pro Sekunde.

Name	Wenn sichtbar	Beschreibung
`ProduceThrottleQueueSize`	Nachdem die Bandbreitenablehnung angewendet wurde.	Die Anzahl der Nachrichten in der Drosselungswarteschlange.
`ProduceThrottleTime`	Nachdem die Bandbreitenablehnung angewendet wurde.	Die Durchschnittszeit der Erzeugungsdrosselung in Millisekunden.
`ProduceTotalTimeMsMean`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die mittlere Erzeugungszeit in Millisekunden.
`RemoteFetchBytesPerSec (RemoteBytesInPerSec in v2.8.2.tiered)`	Nachdem ein Produzent/Verbraucher vorhanden ist.	Die Gesamtzahl der Byte, die als Reaktion auf Verbraucher-Abrufe aus dem gestaffelten Speicher übertragen wurden. Diese Metrik umfasst alle Themenpartitionen, die zum nachgelagerten Transfer-Datenverkehr beitragen. Kategorie: Datenverkehr und Fehlerquoten. Dies ist eine KIP-405-Metrik.
`RemoteCopyBytesPerSec (RemoteBytesOutPerSec in v2.8.2.tiered)`	Nachdem ein Produzent/Verbraucher vorhanden ist.	Die Gesamtzahl der in den gestaffelten Speicher übertragenen Byte, einschließlich Daten aus Protokollsegmenten, Indizes und anderen Hilfsdateien. Diese Metrik umfasst alle Themenpartitionen, die zum nachgelagerten Transfer-Datenverkehr beitragen. Kategorie: Datenverkehr und Fehlerquoten. Dies ist eine KIP-405-Metrik.

Name	Wenn sichtbar	Beschreibung
`RemoteLogManagerTasksAvgIdlePercent`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Der durchschnittliche Prozentsatz der Zeit, die der Remote-Protokoll-Manager im Leerlauf verbracht hat. Der Remote Log Manager überträgt Daten vom Broker in einen gestaffelten Speicher. Kategorie: Interne Aktivität. Dies ist eine KIP-405-Metrik.
`RemoteLogReaderAvgIdlePercent`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Der durchschnittliche Prozentsatz der Zeit, die der Remote-Protokollleser im Leerlauf verbracht hat. Der Remote-Protokollleser überträgt Daten vom Remote-Speicher an den Broker als Reaktion auf Verbraucher-Abrufe. Kategorie: Interne Aktivität. Dies ist eine KIP-405-Metrik.
`RemoteLogReaderTaskQueueSize`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Anzahl der Aufgaben, die für Lesevorgänge aus dem gestaffelten Speicher verantwortlich sind und darauf warten, geplant zu werden. Kategorie: Interne Aktivität. Dies ist eine KIP-405-Metrik.
`RemoteFetchErrorsPerSec (RemoteReadErrorPerSec in v2.8.2.tiered)`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Gesamtfehlerrate bei der Beantwortung von Leseanforderungen, die der angegebene Broker an den gestaffelten Speicher gesendet hat, um Daten als Antwort auf Benutzerabrufe abzurufen. Diese Metrik umfasst alle Themenpartitionen, die zum nachgelagerten Transfer-Datenverkehr beitragen. Kategorie: Datenverkehr und Fehlerquoten. Dies ist eine KIP-405-Metrik.

Name	Wenn sichtbar	Beschreibung
`RemoteFetchRequestsPerSec (RemoteReadRequestsPerSec in v2.8.2.tiered)`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Gesamtzahl der Leseanforderungen, die der angegebene Broker an den gestaffelten Speicher gesendet hat, um Daten als Antwort auf Benutzerabrufe abzurufen. Diese Metrik umfasst alle Themenpartitionen, die zum nachgelagerten Transfer-Datenverkehr beitragen. Kategorie: Datenverkehr und Fehlerquoten. Dies ist eine KIP-405-Metrik.
`RemoteCopyErrorsPerSec (RemoteWriteErrorPerSec in v2.8.2.tiered)`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Gesamtfehlerrate als Antwort auf Schreibanforderungen, die der angegebene Broker zur Übertragung von vorgelagerten Daten an den gestaffelten Speicher gesendet hat. Diese Metrik umfasst alle Themenpartitionen, die zum vorgelagerten Transfer-Datenverkehr beitragen. Kategorie: Datenverkehr und Fehlerquoten. Dies ist eine KIP-405-Metrik.
`ReplicationBytesInPerSec`	Nachdem Sie ein Thema erstellt haben.	Die Anzahl der Bytes, die pro Sekunde von anderen Brokern empfangen werden.
`ReplicationBytesOutPerSec`	Nachdem Sie ein Thema erstellt haben.	Die Anzahl der Bytes, die pro Sekunde an andere Broker gesendet werden.
`RequestExemptFromThrottleTime`	Nachdem die Anforderungsablehnung angewendet wurde.	Die durchschnittliche Zeit (in Millisekunden) für die Verarbeitung der von der Drosselung ausgenommenen Anforderungen in Broker-Netzwerk- und E/A-Threads.

Name	Wenn sichtbar	Beschreibung
`RequestHandlerAvgIdlePercent`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Der durchschnittliche Anteil der Zeit, die sich die Request-Handler-Threads im Leerlauf befinden.
`RequestThrottleQueueSize`	Nachdem die Anforderungsablehnung angewendet wurde.	Die Anzahl der Nachrichten in der Drosselungswarteschlange.
`RequestThrottleTime`	Nachdem die Anforderungsablehnung angewendet wurde.	Die Durchschnittszeit der Anforderungsdrosselung in Millisekunden.
`TcpConnections`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Zeigt die Anzahl der eingehenden und ausgehenden TCP-Segmente an, für die das SYN-Flag gesetzt ist.
`RemoteCopyLagBytes (TotalTierBytesLag in v2.8.2.tiered)`	Nachdem Sie ein Thema erstellt haben.	Die Gesamtzahl der Bytes der Daten, die für die gestaffelte Speicherung auf dem Broker in Frage kommen, aber noch nicht in den gestaffelten Speicher übertragen wurden. Diese Metriken zeigen die Effizienz der vorgelagerten Datenübertragung. Mit zunehmender Verzögerung nimmt die Datenmenge zu, die nicht im gestaffelten Speicher gespeichert wird. Kategorie: Archiv-Verzögerung. Dies ist keine KIP-405-Metrik.
`TrafficBytes`	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Zeigt den Netzwerkverkehr in Gesamtbytes zwischen Clients (Produzenten und Verbrauchern) und Brokern an. Der Verkehr zwischen Brokern wird nicht berichtet.

Name	Wenn sichtbar	Beschreibung
VolumeQueueLength	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Anzahl von Anfragen für Lese- und Schreibvorgänge, die innerhalb eines bestimmten Zeitraums auf Abschluss warten.
VolumeReadBytes	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Anzahl der Bytes, die in einem angegebenen Zeitraum gelesen wurden.
VolumeReadOps	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Anzahl der Lesevorgänge in einem angegebenen Zeitraum.
VolumeTotalReadTime	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Gesamtzahl von Sekunden, die von allen innerhalb eines bestimmte n Zeitraums abgeschlossenen Lesevorgängen aufgewendet wurden.
VolumeTotalWriteTime	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Gesamtzahl von Sekunden, die von allen innerhalb eines bestimmten Zeitraums abgeschlossenen Schreibop erationen aufgewendet wurden.
VolumeWriteBytes	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Anzahl der Bytes, die in einem angegebenen Zeitraum geschrieben wurden.
VolumeWriteOps	Nachdem der Cluster den Status „ACTIVE“ erreicht hat.	Die Gesamtzahl der Schreibvorgänge in einem angegebenen Zeitraum.

Überwachung auf **PER_TOPIC_PER_BROKER**-Ebene

Wenn Sie die Überwachungsebene auf PER_TOPIC_PER_BROKER festlegen, erhalten Sie zusätzlich zu allen in der folgenden Tablette beschriebenen Metriken alle Metriken aus den PER_BROKER und

DEFAULT-Ebenen. Nur die DEFAULT-Ebenenmetriken sind kostenlos. Die Metriken in dieser Tabelle haben die folgenden Dimensionen: Clustername, Broker-ID, Thema.

> **Important**
>
> Für einen Amazon-MSK-Cluster, der Apache Kafka 2.4.1 oder eine neuere Version verwendet, werden die Metriken in der folgenden Tabelle erst angezeigt, nachdem ihre Werte zum ersten Mal ungleich Null sind. Produzenten müssen beispielsweise zuerst Daten an den Cluster senden, um `BytesInPerSec` anzuzeigen.

Zusätzliche Metriken, die ab der **`PER_TOPIC_PER_BROKER`**-Überwachungsebene verfügbar sind

Name	Wenn sichtbar	Beschreibung
`FetchMessageConversionsPerSec`	Nachdem Sie ein Thema erstellt haben.	Die Anzahl der abrufenden Nachrichten, die pro Sekunde konvertiert werden.
`MessagesInPerSec`	Nachdem Sie ein Thema erstellt haben.	Die Anzahl der Nachrichten, die pro Sekunde empfangen werden.
`ProduceMessageConversionsPerSec`	Nachdem Sie ein Thema erstellt haben.	Die Anzahl der Konvertierungen pro Sekunde für produzierte Nachrichten.
`RemoteFetchBytesPerSec (RemoteBytesInPerSec in v2.8.2.tiered)`	Nachdem Sie ein Thema erstellt haben und das Thema produziert/verbraucht.	Die Gesamtzahl der Bytes, die für das angegebene Thema und den angegebenen Broker als Reaktion auf Verbraucher-Abrufe aus dem gestaffelten Speicher übertragen wurden. Diese Metrik umfasst alle Themenpartitionen, die zum nachgelagerten Transfer-Datenverkehr für den angegebenen Broker beitragen. Kategorie: Datenverkehr und Fehlerquoten. Dies ist eine KIP-405-Metrik.
`RemoteCopyBytesPerSec (RemoteBy`	Nachdem Sie ein Thema	Die Anzahl der Bytes, die für das angegebene Thema und den angegebenen Broker in den

Name	Wenn sichtbar	Beschreibung
`tesOutPerSec in v2.8.2.tiered)`	erstellt haben und das Thema produziert/ verbraucht.	gestaffelten Speicher übertragen wurden. Diese Metrik umfasst alle Themenpartitionen, die zum nachgelagerten Transfer-Datenverkehr für den angegebenen Broker beitragen. Kategorie: Datenverkehr und Fehlerquoten. Dies ist eine KIP-405-Metrik.
`RemoteFetchErrorsPerSec (RemoteReadErrorPerSec in v2.8.2.tiered)`	Nachdem Sie ein Thema erstellt haben und das Thema produziert/ verbraucht.	Die Fehlerrate bei der Beantwortung von Leseanforderungen, die der angegebene Broker an den gestaffelten Speicher sendet, um Daten als Antwort auf Benutzerabrufe zum angegebenen Thema abzurufen. Diese Metrik umfasst alle Themenpartitionen, die zum nachgelagerten Transfer-Datenverkehr für den angegebenen Broker beitragen. Kategorie: Datenverkehr und Fehlerquoten. Dies ist eine KIP-405-Metrik.
`RemoteFetchRequestsPerSec (RemoteReadRequestsPerSec in v2.8.2.tiered)`	Nachdem Sie ein Thema erstellt haben und das Thema produziert/ verbraucht.	Die Anzahl der Leseanforderungen, die der angegebene Broker an den gestaffelten Speicher sendet, um Daten als Antwort auf Benutzerabrufe zum angegebenen Thema abzurufen. Diese Metrik umfasst alle Themenpartitionen, die zum nachgelagerten Transfer-Datenverkehr für den angegebenen Broker beitragen. Kategorie: Datenverkehr und Fehlerquoten. Dies ist eine KIP-405-Metrik.

Name	Wenn sichtbar	Beschreibung
`RemoteCopyErrorsPerSec (RemoteWriteErrorPerSec in v2.8.2.tiered)`	Nachdem Sie ein Thema erstellt haben und das Thema produziert/verbraucht.	Die Fehlerrate bei der Beantwortung von Schreibanforderungen, die der angegebene Broker an den gestaffelten Speicher sendet, um Daten in den vorgelagerten Bereich zu übertragen. Diese Metrik umfasst alle Themenpartitionen, die zum nachgelagerten Transfer-Datenverkehr für den angegebenen Broker beitragen. Kategorie: Datenverkehr und Fehlerquoten. Dies ist eine KIP-405-Metrik.

Überwachung auf **`PER_TOPIC_PER_PARTITION`**-Ebene

Wenn Sie die Überwachungsebene auf `PER_TOPIC_PER_PARTITION` festlegen, erhalten Sie zusätzlich zu allen in der folgenden Tablette beschriebenen Metriken alle Metriken aus den `PER_TOPIC_PER_BROKER`-, `PER_BROKER`- und DEFAULT-Ebenen. Nur die `DEFAULT`-Ebenenmetriken sind kostenlos. Die Metriken in dieser Tabelle haben die folgenden Dimensionen: Verbrauchergruppe, Thema, Partition.

Zusätzliche Metriken, die ab der **`PER_TOPIC_PER_PARTITION`**-Überwachungsebene verfügbar sind

Name	Wenn sichtbar	Beschreibung
`EstimatedTimeLag`	Nachdem die Verbrauchergruppe von einem Thema konsumiert hat.	Geschätzte Zeit (in Sekunden), um die Verzögerung beim Partitions-Offset zu verringern.
`OffsetLag`	Nachdem die Verbrauchergruppe von einem Thema konsumiert hat.	Verbraucher-Verzögerung auf Partitionsebene als Anzahl von Offsets.

Amazon MSK-Metriken anzeigen mit CloudWatch

Sie können Metriken für Amazon MSK über die CloudWatch Konsole, die Befehlszeile oder die CloudWatch API überwachen. Die folgenden Verfahren zeigen, wie Sie mithilfe dieser verschiedenen Verfahren auf die Metriken zugreifen können.

So greifen Sie über die Konsole auf Metriken zu CloudWatch

Melden Sie sich bei der an AWS Management Console und öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

1. Wählen Sie im Navigationsbereich Metriken aus.
2. Wählen Sie die Registerkarte Alle Metriken und dann AWS/Kafka.
3. Zum Anzeigen von allgemeinen Metriken auf Themenebene wählen Sie Topic, Broker ID, Cluster Name (Thema, Broker-ID, Cluster-Name), für Metriken auf Broker-Ebene Broker ID, Cluster Name (Broker-ID, Cluster-Name) und für Metriken auf Cluster-Ebene Cluster Name (Cluster-Name) aus.
4. (Optional) Wählen Sie im Grafikbereich eine Statistik und einen Zeitraum aus, und erstellen Sie dann mit diesen Einstellungen einen CloudWatch Alarm.

Um auf Metriken zuzugreifen, verwenden Sie AWS CLI

Verwenden Sie die Befehle list-metrics und get-metric-statistics.

So greifen Sie mit der CloudWatch CLI auf Metriken zu

Verwenden Sie die Befehle mon-list-metrics und mon-get-stats.

Um über die CloudWatch API auf Metriken zuzugreifen

Verwenden Sie die Operationen ListMetricsund GetMetricStatistics.

Überwachung der Verbraucher-Verzögerung

Durch die Überwachung der Verbraucher-Verzögerung können Sie langsame oder festsitzende Verbraucher identifizieren, die nicht mit den neuesten verfügbaren Daten zu einem Thema Schritt halten. Bei Bedarf können Sie dann Abhilfemaßnahmen ergreifen, z. B. diese Verbraucher skalieren oder neu starten. Um die Kundenverzögerung zu überwachen, können Sie Amazon CloudWatch oder Open Monitoring mit Prometheus verwenden.

Metriken zur Verbraucher-Verzögerung quantifizieren den Unterschied zwischen den neuesten Daten, die in Ihren Themen geschrieben wurden, und den Daten, die von Ihren Anwendungen gelesen wurden. Amazon MSK bietet die folgenden Messwerte für Kundenverzögerungen, die Sie über Amazon CloudWatch oder durch offene Überwachung mit Prometheus abrufen können:`EstimatedMaxTimeLag`,,, und`EstimatedTimeLag`. `MaxOffsetLag` `OffsetLag` `SumOffsetLag` Informationen zu diesen Metriken finden Sie unter the section called “Amazon MSK-Metriken für die Überwachung mit CloudWatch”.

> **Note**
>
> Kennzahlen zur Kundenverzögerung sind nur für Verbrauchergruppen sichtbar, die sich im Status STABLE befinden. Eine Verbrauchergruppe ist nach erfolgreichem Abschluss des Rebalancing STABIL, wodurch sichergestellt wird, dass die Partitionen gleichmäßig auf die Verbraucher verteilt sind.

Amazon MSK unterstützt Verbraucher-Verzögerungs-Metriken für Cluster mit Apache Kafka 2.2.1 oder einer späteren Version.

Offene Überwachung mit Prometheus

Sie können Ihren MSK-Cluster mit Prometheus überwachen, einem Open-Source-Überwachungssystem für Zeitreihen-Metrikdaten. Sie können diese Daten mithilfe der Remote-Schreib-Feature von Prometheus in Amazon Managed Service für Prometheus veröffentlichen. Sie können auch Tools verwenden, die mit Prometheus-formatierten Metriken oder Tools die mit Amazon MSK Open Monitoring kompatibel sind, wie etwa Datadog, Lenses, New Relic und Sumo logic. Die offene Überwachung ist kostenlos verfügbar, aber für die Übertragung von Daten über Availability Zones hinweg fallen Gebühren an. Weitere Informationen zu Prometheus finden Sie in der Prometheus-Dokumentation.

Erstellen eines Amazon-MSK-Clusters mit aktivierter offener Überwachung

Unter Verwendung der AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon MSK-Konsole unter https://console.aws.amazon.com/msk/home?region=us-east-1#/home/.
2. Aktivieren Sie unter Monitoring (Überwachung) das Kontrollkästchen neben Enable open monitoring with Prometheus (Offene Überwachung mit Prometheus aktivieren).

3. Geben Sie die erforderlichen Informationen in allen Abschnitten der Seite an und überprüfen Sie die verfügbaren Optionen.
4. Wählen Sie Cluster erstellen.

Verwenden Sie den AWS CLI

- Rufen Sie den Befehl create-cluster auf und geben Sie die Option `open-monitoring` an. Aktivieren Sie `JmxExporter`, `NodeExporter` oder beides. Wenn Sie `open-monitoring` angeben, können die beiden Exporteure nicht gleichzeitig deaktiviert werden.

Verwenden der API

- Rufen Sie den CreateClusterVorgang auf und geben Sie an`OpenMonitoring`. Aktivieren Sie `jmxExporter`, `nodeExporter` oder beides. Wenn Sie `OpenMonitoring` angeben, können die beiden Exporteure nicht gleichzeitig deaktiviert werden.

Aktivieren der offenen Überwachung für einen vorhandenen Amazon-MSK-Cluster

Um die offene Überwachung zu aktivieren, stellen Sie sicher, dass sich der Cluster im Status `ACTIVE` befindet.

Mit dem AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon MSK-Konsole unter https://console.aws.amazon.com/msk/home?region=us-east-1#/home/.
2. Wählen Sie den Namen des Clusters, den Sie aktualisieren möchten. Dadurch gelangen Sie zu einer Seite mit Details für den Cluster.
3. Scrollen Sie auf der Registerkarte Eigenschaften nach unten zum Abschnitt Überwachung.
4. Wählen Sie Bearbeiten aus.
5. Aktivieren Sie das Kontrollkästchen neben Enable open monitoring with Prometheus (Offene Überwachung mit Prometheus aktivieren).
6. Wählen Sie Änderungen speichern aus.

Verwenden Sie den AWS CLI

- Rufen Sie den Befehl update-monitoring auf und geben Sie die Option `open-monitoring` an. Aktivieren Sie `JmxExporter`, `NodeExporter` oder beides. Wenn Sie `open-monitoring` angeben, können die beiden Exporteure nicht gleichzeitig deaktiviert werden.

Verwenden der API

- Rufen Sie den UpdateMonitoringVorgang auf und geben Sie an`OpenMonitoring`. Aktivieren Sie `jmxExporter`, `nodeExporter` oder beides. Wenn Sie `OpenMonitoring` angeben, können die beiden Exporteure nicht gleichzeitig deaktiviert werden.

Einrichten eines Prometheus-Hosts auf einer Amazon-EC2-Instance

1. Laden Sie den Prometheus-Server von https://prometheus.io/download/#prometheus auf Ihre Amazon-EC2-Instance herunter.
2. Extrahieren Sie die heruntergeladene Datei in ein Verzeichnis und navigieren Sie zu diesem Verzeichnis.
3. Erstellen Sie eine Datei mit dem folgenden Inhalt und geben Sie ihr den Namen `prometheus.yml`.

```
# 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. Verwenden Sie den ListNodesVorgang, um eine Liste der Broker Ihres Clusters abzurufen.
5. Erstellen Sie eine Datei namens `targets.json` mit dem folgenden JSON: Ersetzen Sie *broker_dns_1*, *broker_dns_2* und den Rest des Broker-DNS-Namen durch die DNS-Namen, die Sie im vorherigen Schritt für Ihre Broker erhalten haben. Geben Sie alle Broker an, die Sie im vorherigen Schritt erhalten haben. Amazon MSK verwendet Port 11001 für den JMX Exporter und Port 11002 für den 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**
>
> Um JMX-Metriken von KraFT-Controllern zu entfernen, fügen Sie der JSON-Datei Controller-DNS-Namen als Ziele hinzu. Zum Beispiel: `controller_dns_1:11001` durch den tatsächlichen `controller_dns_1` DNS-Namen des Controllers ersetzen.

6. Um den Prometheus-Server auf Ihrer Amazon-EC2-Instance zu starten, führen Sie den folgenden Befehl in dem Verzeichnis aus, in dem Sie die Prometheus-Dateien extrahiert und `prometheus.yml` und `targets.json` gespeichert haben.

```
./prometheus
```

7. Suchen Sie die öffentliche IPv4-IP-Adresse der Amazon-EC2-Instance, auf der Sie Prometheus im vorherigen Schritt ausgeführt haben. Sie benötigen diese öffentliche IP-Adresse im folgenden Schritt.
8. Um auf die Prometheus-Web-UI zuzugreifen, öffnen Sie einen Browser, der auf Ihre Amazon-EC2-Instance zugreifen kann, und navigieren Sie zu *Prometheus-Instance-Public-IP*`:9090`, wobei *Prometheus-Instance-Public-IP* die öffentliche IP-Adresse ist, die Sie im vorherigen Schritt erhalten haben.

Prometheus-Metriken

Alle von Apache Kafka an JMX ausgegebenen Metriken sind über eine offene Überwachung mit Prometheus zugänglich. Informationen zu Apache Kafka-Metriken finden Sie unter Monitoring in der Apache Kafka-Dokumentation. Neben Apache-Kafka-Metriken sind auch Verbraucher-Verzögerungs-Metriken auf Port 11001 unter dem JMX-MBean-Namen `kafka.consumer.group:type=ConsumerLagMetrics` verfügbar. Sie können auch den Prometheus Node Exporter verwenden, um CPU- und Festplattenmetriken für Ihre Broker von Port 11002 abzurufen.

Speichern von Prometheus-Metriken in Amazon Managed Service für Prometheus

Amazon Managed Service for Prometheus ist ein Prometheus-kompatibler Service zur Überwachung und Warnung, den Sie zur Überwachung von Amazon-MSK-Clustern verwenden können. Es ist ein vollständig verwalteter Service, der die Aufnahme, Speicherung, Abfrage und Warnung Ihrer Metriken automatisch skaliert. Es lässt sich auch in AWS Sicherheitsdienste integrieren, um Ihnen einen schnellen und sicheren Zugriff auf Ihre Daten zu ermöglichen. Sie können die Open-Source-PromQL-Abfragesprache verwenden, um Ihre Metriken abzufragen und darauf zu warnen.

Weitere Informationen finden Sie unter Erste Schritte mit Amazon Managed Service for Prometheus.

Amazon-MSK-Speicherkapazitätswarnungen

Auf von Amazon MSK bereitgestellten Clustern wählen Sie die primäre Speicherkapazität des Clusters aus. Wenn Sie die Speicherkapazität eines Brokers in Ihrem bereitgestellten Cluster

ausschöpfen, kann sich dies auf dessen Fähigkeit auswirken, Daten zu produzieren und zu nutzen, was zu kostspieligen Ausfallzeiten führen kann. Amazon MSK bietet CloudWatch Metriken, mit denen Sie die Speicherkapazität Ihres Clusters überwachen können. Um Ihnen das Erkennen und Beheben von Speicherkapazitätsproblemen zu erleichtern, sendet Ihnen Amazon MSK jedoch automatisch dynamische Cluster-Speicherkapazitätswarnungen. Die Speicherkapazitätswarnungen enthalten Empfehlungen für kurzfristige und langfristige Schritte zur Verwaltung der Speicherkapazität Ihres Clusters. Von der Amazon-MSK-Konsole aus können Sie Quicklinks in den Benachrichtigungen verwenden, um sofort empfohlene Maßnahmen zu ergreifen.

Es gibt zwei Arten von MSK-Warnmeldungen zur Speicherkapazität: proaktive Benachrichtigungen und Warnmeldungen zur Behebung von Problemen.

- Proaktive („Aktion erforderlich“) Warnmeldungen zur Speicherkapazität warnen Sie vor potenziellen Speicherproblemen in Ihrem Cluster. Wenn ein Broker in einem MSK-Cluster mehr als 60 oder 80 % seiner Festplattenspeicherkapazität genutzt hat, erhalten Sie proaktive Benachrichtigungen zum betroffenen Broker.
- Bei Warnmeldungen zur Behebung der Speicherkapazität („Kritische Aktion erforderlich“) müssen Sie Abhilfemaßnahmen ergreifen, um ein kritisches Clusterproblem zu beheben, wenn einer der Broker in Ihrem MSK-Cluster über keine Festplattenspeicherkapazität mehr verfügt.

Amazon MSK sendet diese Benachrichtigungen automatisch an die Amazon MSK-Konsole, AWS Health Dashboard EventBridge, Amazon und E-Mail-Kontakte für Ihr AWS Konto. Sie können Amazon auch so konfigurieren EventBridge, dass diese Benachrichtigungen an Slack oder an Tools wie New Relic und Datadog gesendet werden.

Warnmeldungen zur Speicherkapazität sind standardmäßig für alle von MSK bereitgestellten Cluster aktiviert und können nicht deaktiviert werden. Dieses Feature ist in allen Regionen verfügbar, in denen MSK verfügbar ist.

Überwachen der Speicherkapazitätswarnungen in Amazon MSK

Sie können auf verschiedene Arten nach Warnmeldungen zur Speicherkapazität suchen:

- Rufen Sie die Amazon-MSK-Konsole auf. Warnungen zur Speicherkapazität werden 90 Tage lang im Bereich „Cluster alerts“ (Clusterwarnungen) angezeigt. Die Warnmeldungen enthalten Empfehlungen und Einfachklick-Linkaktionen, um Probleme mit der Festplattenspeicherkapazität zu beheben.

- Verwenden Sie ListClustersdie APIs ListClustersV2 oder DescribeClusterV2 DescribeCluster, um alle Benachrichtigungen für einen Cluster anzuzeigen`CustomerActionStatus`.
- Gehen Sie zum AWS Health Dashboard, um Benachrichtigungen von MSK und anderen AWS Diensten anzuzeigen.
- Richten Sie AWS Health API und Amazon ein EventBridge, um Warnmeldungen an Plattformen von Drittanbietern wie Datadog und Slack NewRelic weiterzuleiten.

Verwenden von LinkedIn's Cruise Control für Apache Kafka mit Amazon MSK

Sie können den Tempomat verwenden LinkedIn, um Ihren Amazon MSK-Cluster neu auszurichten, Anomalien zu erkennen und zu beheben und den Status und den Zustand des Clusters zu überwachen.

So können Sie Cruise Control herunterladen und einrichten

1. Erstellen Sie in derselben Amazon VPC wie der Amazon-MSK-Cluster eine Amazon-EC2-Instance.
2. Installieren Sie Prometheus auf der Amazon-EC2-Instance, die Sie im vorherigen Schritt erstellt haben. Notieren Sie sich die private IP und den Port. Die Standard-Portnummer ist 9090. Weitere Informationen zur Konfiguration von Prometheus zum Aggregieren von Metriken für Ihren Cluster finden Sie unter the section called “Offene Überwachung mit Prometheus”.
3. Laden Sie Cruise Control auf die Amazon-EC2-Instance herunter. (Alternativ können Sie eine separate Amazon-EC2-Instance für Cruise Control verwenden, wenn Sie dies bevorzugen.) Verwenden Sie für einen Cluster mit Apache Kafka Version 2.4.* die neueste Version 2.4.* von Cruise Control. Wenn Ihr Cluster über eine Apache-Kafka-Version verfügt, die älter als 2.4.* ist, verwenden Sie die neueste Version 2.0.* von Cruise Control.
4. Dekomprimieren Sie die Cruise-Control-Datei und wechseln Sie dann in den dekomprimierten Ordner.
5. Führen Sie zum Installieren von git den folgenden Befehl aus.

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

6. Führen Sie den folgenden Befehl aus, um das lokale Repository zu initialisieren. Ersetzen Sie *Your-Cruise-Control-Folder* durch den Namen Ihres aktuellen Ordners (den Ordner, den Sie beim Dekomprimieren des Cruise-Control-Downloads erhalten haben).

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

7. Führen Sie den folgenden Befehl zum Entwickeln des Quell-Codes aus.

   ```
   ./gradlew jar copyDependantLibs
   ```

So können Sie Cruise Control konfigurieren und ausführen

1. Nehmen Sie die folgenden Änderungen an der Datei `config/cruisecontrol.properties` vor. Ersetzen Sie die Beispiel-Bootstrap-Server und die Bootstrap-Brokers-Zeichenfolge durch die Werte für Ihren Cluster. Um diese Zeichenfolgen für Ihren Cluster abzurufen, können Sie sich die Cluster-Details in der Konsole ansehen. Alternativ können Sie die DescribeClusterAPI-Operationen GetBootstrapBrokersund oder deren CLI-Entsprechungen verwenden.

```
# 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
```

2. Bearbeiten Sie die `config/capacityCores.json`-Datei, um die richtige Festplattengröße und die richtigen CPU-Kerne sowie die Netzwerk-Ein-/Ausgangsgrenzen anzugeben. Sie können den DescribeClusterAPI-Vorgang (oder sein CLI-Äquivalent) verwenden, um die Festplattengröße zu ermitteln. Informationen zu CPU-Kernen und Netzwerkbeschränkungen finden Sie unter Amazon-EC2-Instance-Typen.

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

3. Sie können optional die Cruise-Control-Benutzeroberfläche installieren. Um es herunterzuladen, wechseln Sie zu Einrichten des Cruise-Control-Frontend.
4. Führen Sie den folgenden Befehl aus, um Cruise Control zu starten. Erwägen Sie, ein Tool wie `screen` oder `tmux` zu verwenden, um eine Sitzung mit langer Laufzeit offen zu halten.

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

5. Stellen Sie mithilfe der Cruise-Control-APIs oder der Benutzeroberfläche sicher, dass Cruise Control über die Cluster-Lastdaten verfügt und Vorschläge zum Neuausgleich macht. Es kann einige Minuten dauern, bis ein gültiges Metrikfenster angezeigt wird.

Automatisierte Bereitstellungsvorlage von Cruise Control für Amazon MSK

Sie können diese CloudFormation Vorlage auch verwenden, um Cruise Control und Prometheus einfach bereitzustellen, um tiefere Einblicke in die Leistung Ihres Amazon MSK-Clusters zu erhalten und die Ressourcennutzung zu optimieren.

Wichtigste Funktionen:

- Automatisierte Bereitstellung einer Amazon EC2 EC2-Instance mit vorkonfiguriertem Cruise Control und Prometheus.
- Support für von Amazon MSK bereitgestellte Cluster.
- Flexible Authentifizierung mit PlainText und IAM.
- Cruise Control ist nicht von Zookeeper abhängig.
- Passen Sie Prometheus-Ziele, Cruise Control-Kapazitätseinstellungen und andere Konfigurationen ganz einfach an, indem Sie Ihre eigenen Konfigurationsdateien bereitstellen, die in einem Amazon S3 S3-Bucket gespeichert sind.

Amazon-MSK-Kontingent

Ihr AWS Konto hat Standardkontingente für Amazon MSK. Sofern nicht anders angegeben, ist jedes Kontingent pro Konto innerhalb Ihres Kontos regionsspezifisch. AWS

Amazon-MSK-Kontingent

- Bis zu 90 Broker pro Konto. 30 Broker pro ZooKeeper Modus-Cluster. 60 Broker pro KraFT-Modus-Cluster. Um ein höheres Kontingent anzufordern, rufen Sie das Support Center der AWS Konsole auf und erstellen Sie einen Support-Fall.
- Mindestens 1 GiB Speicher pro Broker.
- Maximal 16384 GiB Speicher pro Broker.
- Ein Cluster, der the section called “IAM-Zugriffssteuerung” verwendet, kann zu einem beliebigen Zeitpunkt bis zu 3 000 TCP-Verbindungen pro Broker haben. Um dieses Limit zu erhöhen, können Sie die Konfigurationseigenschaft `listener.name.client_iam.max.connections` oder die `listener.name.client_iam_public.max.connections` Konfigurationseigenschaft mithilfe der AlterConfig Kafka-API oder des `kafka-configs.sh` Tools anpassen. Es ist wichtig zu beachten, dass das Erhöhen einer der beiden Eigenschaften auf einen hohen Wert die Verfügbarkeit beeinträchtigen kann.
- Beschränkungen für TCP-Verbindungen. Wenn Verbindungsraten-Bursts aktiviert sind, erlaubt MSK 100 Verbindungen pro Sekunde. Die Ausnahme ist der Instance-Typ kafka.t3.small, für den 4 Verbindungen pro Sekunde bei aktivierten Verbindungsraten-Bursts zulässig sind. Bei älteren Clustern, für die Verbindungsraten-Bursts nicht aktiviert sind, wird die Funktion automatisch aktiviert, wenn der Cluster gepatcht wird.

  Um Wiederholungsversuche bei fehlgeschlagenen Verbindungen zu verarbeiten, können Sie den Konfigurationsparameter `reconnect.backoff.ms` auf der Client-Seite festlegen. Wenn Sie beispielsweise möchten, dass ein Client Verbindungen nach 1 Sekunde erneut versucht, legen Sie `reconnect.backoff.ms` auf 1 000 fest. Weitere Informationen finden Sie unter reconnect.backoff.ms in der Apache-Kafka-Dokumentation.
- Bis zu 100 Konfigurationen pro Konto. Um ein höheres Kontingent anzufordern, rufen Sie das Support-Center der AWS -Konsole auf und erstellen Sie einen Support-Fall.
- Maximal 50 Revisionen pro Konfiguration.
- Um die Konfiguration oder die Apache-Kafka-Version eines MSK-Clusters zu aktualisieren, stellen Sie zunächst sicher, dass die Anzahl der Partitionen pro Broker unter den in the section called “ Die

Größe Ihres Clusters anpassen: Anzahl der Partitionen pro Broker" beschriebenen Grenzwerten liegt.

MSK Replicator-Kontingente

- Maximal 15 MSK-Replikatoren pro Konto.
- MSK Replicator repliziert nur bis zu 750 Themen in sortierter Reihenfolge. Wenn Sie mehr Themen replizieren müssen, empfehlen wir Ihnen, einen separaten Replicator zu erstellen. Rufen Sie das AWS Konsolen-Supportcenter auf und erstellen Sie einen Support-Fall, wenn Sie Support für mehr als 750 Themen pro Replicator benötigen. Sie können die Anzahl der replizierten Themen mithilfe der Metrik "TopicCount" überwachen.
- Ein maximaler Eingangsdurchsatz von 1 GB pro Sekunde pro MSK-Replikator. Um ein höheres Kontingent anzufordern, rufen Sie das Support Center der AWS Konsole auf und erstellen Sie einen Support-Fall.
- MSK Replicator-Datensatzgröße — Maximal 10 MB Datensatzgröße (message.max.bytes). Um ein höheres Kontingent anzufordern, rufen Sie das Support Center der AWS Konsole auf und erstellen Sie einen Support-Fall.

MSK-Serverless-Kontingent

> **Note**
>
> Wenn Sie Probleme mit den Kontingentbeschränkungen haben, wenden Sie sich an den AWS Support, indem Sie eine Support-Anfrage erstellen.

Die Limits gelten pro Cluster, sofern nicht anders angegeben.

Dimension	Kontingent	Ergebnis einer Kontingentverletzung
Maximaler Eingangsdurchsatz	200 Mbit/s	Verlangsamung mit Drosselungsdauer als Reaktion
Maximaler Eingangsdurchsatz	400 Mbit/s	Verlangsamung mit Drosselungsdauer als Reaktion

Dimension	Kontingent	Ergebnis einer Kontingentverletzung
Maximale Aufbewahrungsdauer	Unbegrenzt	N/A
Maximale Anzahl von Client-Verbindungen	3000	Verbindung geschlossen
Maximale Verbindungsversuche	100 pro Sekunde	Verbindung geschlossen
Maximale Nachrichtengröße	8 MB	Die Anfrage schlägt fehl mit ErrorCode: INVALID_REQUEST
Maximale Anforderungsrate	15 000 pro Sekunde	Verlangsamung mit Drosselungsdauer als Reaktion
Maximale Rate von Anfragen an Themen-Management-APIs	2 pro Sekunde	Verlangsamung mit Drosselungsdauer als Reaktion
Maximale Anzahl an abrufbaren Bytes pro Anfrage	55 MB	Die Anfrage schlägt fehl mit ErrorCode: INVALID_REQUEST
Maximale Anzahl von Verbrauchergruppen	500	JoinGroup Anfrage schlägt fehl
Maximale Anzahl von Partitionen (Leader)	2 400 für nicht komprimierte Themen. 120 für komprimierte Themen. Um eine Kontingentanpassung anzufordern, rufen Sie das Support Center für die AWS Konsole auf und erstellen Sie eine Support-Anfrage.	Die Anfrage schlägt fehl mit ErrorCode: INVALID_REQUEST

Dimension	Kontingent	Ergebnis einer Kontingentverletzung
Maximale Geschwindigkeit beim Erstellen und Löschen von Partitionen	250 in 5 Minuten	Die Anfrage schlägt fehl mit ErrorCode: THROUGHPUT_QUOTA_EXCEEDEED
Maximaler Eingangsdurchsatz pro Partition	5 Mbit/s	Verlangsamung mit Drosselungsdauer als Reaktion
Maximaler Ausgangsdurchsatz pro Partition	10 Mbit/s	Verlangsamung mit Drosselungsdauer als Reaktion
Maximale Partitionsgröße (für komprimierte Themen)	250 GB	Die Anfrage schlägt fehl mit: THROUGHPUT_QUOTA_EXCEEDEED ErrorCode
Maximale Anzahl von Client-VPCs pro Serverless-Cluster	5	
Maximale Anzahl von Serverless-Clustern pro Konto	10. Um eine Kontingentanpassung anzufordern, rufen Sie das Support Center für die AWS Konsole auf und erstellen Sie eine Support-Anfrage.	

MSK-Connect-Kontingent

- Bis zu 100 benutzerdefinierte Plugins.
- Bis zu 100 Worker-Konfigurationen.
- Bis zu 60 Connect-Worker. Wenn ein Konnektor für automatisch skalierte Kapazität eingerichtet ist, verwendet MSK Connect die maximale Anzahl von Workern, die für diesen Konnektor konfiguriert sind, um das Kontingent für das Konto zu berechnen.
- Bis zu 10 Worker pro Anschluss.

Um ein höheres Kontingent für MSK Connect anzufordern, rufen Sie das AWS Konsolen-Supportcenter auf und erstellen Sie einen Support-Fall.

Amazon-MSK-Ressourcen

Der Begriff Ressourcen hat in Amazon MSK je nach Kontext zwei Bedeutungen. Im Kontext von APIs ist eine Ressource eine Struktur, mit der Sie einen Vorgang aufrufen können. Eine Liste dieser Ressourcen und der Vorgänge, die Sie für sie aufrufen können, finden Sie unter Ressourcen in der API-Referenz zu Amazon MSK. Im Kontext von the section called “IAM-Zugriffssteuerung” ist eine Ressource eine Entität, für die Sie den Zugriff gewähren oder verweigern können, wie im Abschnitt the section called “Ressourcen” definiert.

MSK-Integrationen

Dieser Abschnitt enthält Verweise auf AWS Funktionen, die in Amazon MSK integriert sind.

Themen

Amazon-Athena-Konnektor für Amazon MSK

Der Amazon-Athena-Konnektor für Amazon MSK ermöglicht es Amazon Athena, SQL-Abfragen für Apache-Kafka-Themen auszuführen. Verwenden Sie diesen Konnektor, um Apache-Kafka-Themen als Tabellen und Nachrichten als Zeilen in Athena anzuzeigen.

Weitere Informationen finden Sie unter Amazon Athena MSK Connector im Benutzerhandbuch für Amazon Athena.

Amazon-Redshift-Streaming-Datenerfassung

Amazon Redshift unterstützt die Streaming-Erfassung von Amazon MSK. Die Streaming-Erfassungs-Feature von Amazon Redshift ermöglicht das Erfassen von Streaming-Daten mit geringer Latenz und hoher Geschwindigkeit aus Amazon MSK in einer materialisierten Ansicht von Amazon Redshift. Da keine Daten in Amazon S3 bereitgestellt werden müssen, kann Amazon Redshift Streaming-Daten mit geringerer Latenz und geringeren Speicherkosten erfassen. Sie können die Amazon-Redshift-Streaming-Erfassung auf einem Amazon-Redshift-Cluster mithilfe von SQL-Anweisungen konfigurieren, um sich zu authentifizieren und eine Verbindung zu einem Amazon-MSK-Thema herzustellen.

Weitere Informationen finden Sie unter Streaming-Erfassung im Entwicklerhandbuch für Amazon Redshift Database.

Firehose

Amazon MSK ist in Firehose integriert, um eine serverlose, codefreie Lösung für die Übertragung von Streams von Apache Kafka-Clustern an Amazon S3 S3-Datenseen bereitzustellen. Firehose ist ein Streaming-Dienst zum Extrahieren, Transformieren und Laden (ETL), der Daten aus Ihren Amazon MSK-Kafka-Themen liest, Transformationen wie die Konvertierung in Parquet durchführt und die Daten aggregiert und in Amazon S3 schreibt. Mit wenigen Klicks von der Konsole aus können Sie einen Firehose-Stream einrichten, um aus einem Kafka-Thema zu lesen und an einen S3-Standort zu liefern. Es muss kein Code geschrieben werden, es gibt keine Konnektor-Anwendungen und es müssen keine Ressourcen bereitgestellt werden. Firehose skaliert automatisch auf der Grundlage der zum Kafka-Thema veröffentlichten Datenmenge, und Sie zahlen nur für die von Kafka aufgenommenen Bytes.

Weitere Informationen über dieses Feature finden Sie im Folgenden.

- Mit Amazon MSK in Kinesis Data Firehose schreiben — Amazon Kinesis Data Firehose im Amazon Data Firehose Developer Guide
- Blog: Amazon MSK stellt Ihrem Data Lake Managed Data Delivery von Apache Kafka vor
- Lab: Lieferung an Amazon S3 mit Firehose

Zugriff auf Amazon EventBridge Pipes über die Amazon MSK-Konsole

Amazon EventBridge Pipes verbindet Quellen mit Zielen. Pipes sind für point-to-point Integrationen zwischen unterstützten Quellen und Zielen vorgesehen und unterstützen erweiterte Transformationen und Anreicherungen. EventBridge Pipes bieten eine hoch skalierbare Möglichkeit, Ihren Amazon MSK-Cluster mit AWS Services wie Step Functions, Amazon SQS und API Gateway sowie Software-as-a-Service (SaaS) -Anwendungen von Drittanbietern wie Salesforce zu verbinden.

Um eine Pipe einzurichten, wählen Sie die Quelle aus, fügen Sie optionale Filterung hinzu, definieren Sie die optionale Anreicherung und wählen Sie das Ziel für die Ereignisdaten.

Auf der Detailseite für einen Amazon-MSK-Cluster können Sie die Pipes anzeigen, die diesen Cluster als Quelle verwenden. Von dort aus können Sie auch:

- Starten Sie die EventBridge Konsole, um die Pipe-Details anzuzeigen.
- Starten Sie die EventBridge Konsole, um eine neue Pipe mit dem Cluster als Quelle zu erstellen.

Weitere Informationen zur Konfiguration eines Amazon MSK-Clusters als Pipe-Quelle finden Sie unter Amazon Managed Streaming for Apache Kafka Cluster as a source im EventBridge Amazon-Benutzerhandbuch. Weitere Informationen zu EventBridge Pipes im Allgemeinen finden Sie unter EventBridge Pipes.

So greifen Sie auf EventBridge Pipes für einen bestimmten Amazon MSK-Cluster zu

1. Öffnen Sie die Amazon-MSK-Konsole und wählen Sie dann Cluster.
2. Wählen Sie einen Cluster aus.
3. Wählen Sie auf der Seite der Cluster-Details die Registerkarte Integration.

   Die Registerkarte Integration enthält eine Liste aller Pipes, die derzeit für die Verwendung des ausgewählten Clusters als Quelle konfiguriert sind, darunter:

   - Pipe-Name
   - aktueller Status
   - Pipe-Ziel
   - wann die Pipe zuletzt geändert wurde
4. Verwalten Sie die Pipes für Ihren Amazon-MSK-Cluster wie gewünscht:

   So greifen Sie auf weitere Details zu einer Pipe zu

   - Wählen Sie die Pipe.

     Dadurch wird die Seite mit den Pipe-Details der EventBridge Konsole geöffnet.

   So erstellen Sie eine neue Pipe

   - Wählen Sie Amazon-MSK-Cluster mit Pipe verbinden.

     Dadurch wird die Seite „Pipe erstellen“ der EventBridge Konsole geöffnet, auf der der Amazon MSK-Cluster als Pipe-Quelle angegeben ist. Weitere Informationen finden Sie unter Erstellen einer EventBridge Pipe im EventBridge Amazon-Benutzerhandbuch.

   - Sie können auch auf der Clusters-Seite eine Pipe für einen Cluster erstellen. Wählen Sie den Cluster aus und wählen Sie im Menü Aktionen die Option EventBridge Pipe erstellen aus.

Apache-Kafka-Versionen

Wenn Sie einen Amazon-MSK-Cluster erstellen, geben Sie an, welche Apache-Kafka-Version Sie darauf ausführen möchten. Sie können auch die Apache Kafka-Version eines vorhandenen Clusters aktualisieren. Die Themen in diesem Kapitel helfen Ihnen, die Zeitpläne für die Unterstützung der Kafka-Version sowie Vorschläge für bewährte Verfahren zu verstehen.

Themen

Unterstützte Apache Kafka-Versionen

Amazon Managed Streaming für Apache Kafka (Amazon MSK) unterstützt die folgenden Versionen von Apache Kafka und Amazon MSK. Die Apache Kafka-Community bietet etwa 12 Monate Support für eine Version nach dem Veröffentlichungsdatum. Weitere Informationen finden Sie in der Apache Kafka EOL-Richtlinie (End of Life).

Unterstützte Apache Kafka-Versionen

Apache Kafka-Version	Veröffentlichungsdatum von MSK	Datum des Endes des Supports
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
2.4.1	02.04.2020	2024-06-08
2.4.1.1	09.09.2020	2024-06-08
2,5.1	30.09.2020	2024-06-08
2,6,0	21.10.2020	2024-09-11

Apache Kafka-Version	Veröffentlichungsdatum von MSK	Datum des Endes des Supports
2.6.1	19.01.2021	2024-09-11
2.6.2	29.04.2021	2024-09-11
2.6.3	21.12.2021	2024-09-11
2,7,0	29.12.2020	2024-09-11
2.7.1	25.05.2021	11. September 2024
2.7.2	21.12.2021	2024-09-11
2,8,0	--	2024-09-11
2,8.1	28.10.2022	2024-09-11
2.8.2 gestaffelt	28.10.2022	Wird noch bekannt gegeben
3.1.1	22.06.2022	11. September 2024
3.2.0	22.06.2022	11. September 2024
3.3.1	26.10.2022	2024-09-11
3.3.2	2023-03-02	2024-09-11
3,4,0	2023-05-04	2025-06-17
3.5.1 (empfohlen)	2023-09-26	--
3,6,0	16.11.2023-23	--
3.7.x	29.05.2024	--

Weitere Informationen zur Support-Richtlinie für Amazon MSK-Versionen finden Sie unterSupport-Richtlinie für Amazon MSK-Versionen.

Apache Kafka Version 3.7.x (mit produktionsbereitem Tiered Storage)

Apache Kafka Version 3.7.x auf MSK beinhaltet Unterstützung für Apache Kafka Version 3.7.0. Sie können Cluster erstellen oder bestehende Cluster aktualisieren, um die neue Version 3.7.x zu verwenden. Mit dieser Änderung der Versionsbezeichnung müssen Sie keine neueren Patchfix-Versionen wie 3.7.1 mehr verwenden, wenn sie von der Apache Kafka-Community veröffentlicht werden. Amazon MSK aktualisiert 3.7.x automatisch, um future Patch-Versionen zu unterstützen, sobald diese verfügbar sind. Auf diese Weise können Sie von den Sicherheits- und Bugfixes profitieren, die über Patchfix-Versionen verfügbar sind, ohne ein Versions-Upgrade auszulösen. Diese von Apache Kafka veröffentlichten Patchfix-Versionen beeinträchtigen nicht die Versionskompatibilität, und Sie können von den neuen Patchfix-Versionen profitieren, ohne sich Gedanken über Lese- oder Schreibfehler Ihrer Client-Anwendungen machen zu müssen. Bitte stellen Sie sicher, dass Ihre Tools zur Infrastrukturautomatisierung, wie z. B. CloudFormation, aktualisiert sind, um dieser Änderung der Versionsbezeichnung Rechnung zu tragen.

Amazon MSK unterstützt jetzt den Kraft-Modus (Apache Kafka Raft) in Apache Kafka Version 3.7.x. Bei Amazon MSK sind Kraft-Controller wie bei ZooKeeper Nodes ohne zusätzliche Kosten für Sie enthalten und erfordern keine zusätzliche Einrichtung oder Verwaltung durch Sie. Sie können jetzt Cluster entweder im KraFT-Modus oder im ZooKeeper Modus auf Apache Kafka Version 3.7.x erstellen. Im Kraft-Modus können Sie bis zu 60 Broker hinzufügen, um mehr Partitionen pro Cluster zu hosten, ohne eine Erhöhung des Limits zu beantragen, verglichen mit dem Kontingent von 30 Brokern bei Zookeeper-basierten Clustern. Weitere Informationen zu KraFT auf MSK finden Sie unter KraFt-Modus.

Apache Kafka Version 3.7.x enthält auch mehrere Bugfixes und neue Funktionen, die die Leistung verbessern. Zu den wichtigsten Verbesserungen gehören Leader-Discovery-Optimierungen für Clients und Optionen zur Optimierung des Log-Segment-Flushs. Eine vollständige Liste der Verbesserungen und Bugfixes finden Sie in den Apache Kafka-Versionshinweisen für 3.7.0.

Apache Kafka Version 3.6.0 (mit produktionsbereiter gestaffelter Speicherung)

Weitere Informationen zu Apache Kafka Version 3.6.0 (mit produktionsbereiter gestaffelter speicherung) finden Sie in den Versionshinweisen auf der Download-Seite von Apache Kafka.

Amazon MSK wird in dieser Version aus Stabilitätsgründen weiterhin Zookeeper für die Quorumverwaltung verwenden und verwalten.

Amazon MSK versie 3.5.1

Amazon Managed Streaming for Apache Kafka (Amazon MSK) unterstützt jetzt Apache Kafka Version 3.5.1 für neue und bestehende Cluster. Apache Kafka 3.5.1 enthält mehrere Bugfixes und neue Funktionen, die die Leistung verbessern. Zu den wichtigsten Funktionen gehört die Einführung einer neuen Rack-fähigen Partitionszuweisung für Privatanwender. Amazon MSK wird in dieser Version weiterhin Zookeeper für die Quorumverwaltung verwenden und verwalten. Eine vollständige Liste der Verbesserungen und Bugfixes finden Sie in den Apache Kafka-Versionshinweisen für 3.5.1.

Weitere Informationen zu Apache Kafka Version 3.5.1 finden Sie in den Versionshinweisen auf der Download-Seite von Apache Kafka.

Amazon MSK versie 3.4.0

Amazon Managed Streaming for Apache Kafka (Amazon MSK) unterstützt jetzt Apache Kafka Version 3.4.0 für neue und bestehende Cluster. Apache Kafka 3.4.0 enthält mehrere Bugfixes und neue Funktionen, die die Leistung verbessern. Zu den wichtigsten Funktionen gehört ein Fix zur Verbesserung der Stabilität beim Abrufen aus dem nächstgelegenen Replikat. Amazon MSK wird in dieser Version weiterhin Zookeeper für die Quorumverwaltung verwenden und verwalten. Eine vollständige Liste der Verbesserungen und Bugfixes finden Sie in den Apache Kafka-Versionshinweisen für 3.4.0.

Weitere Informationen zu Apache Kafka Version 3.4.0 finden Sie in den Versionshinweisen auf der Download-Seite von Apache Kafka.

Amazon MSK versie 3.3.2

Amazon Managed Streaming for Apache Kafka (Amazon MSK) unterstützt jetzt Apache Kafka Version 3.3.2 für neue und bestehende Cluster. Apache Kafka 3.3.2 enthält mehrere Bugfixes und neue Funktionen, die die Leistung verbessern. Zu den wichtigsten Funktionen gehört ein Fix zur Verbesserung der Stabilität beim Abrufen aus dem nächstgelegenen Replikat. Amazon MSK wird in dieser Version weiterhin Zookeeper für die Quorumverwaltung verwenden und verwalten. Eine vollständige Liste der Verbesserungen und Bugfixes finden Sie in den Apache Kafka-Versionshinweisen für 3.3.2.

Weitere Informationen zu Apache Kafka Version 3.3.2 finden Sie in den Versionshinweisen auf der Download-Seite von Apache Kafka.

Amazon MSK versie 3.3.1

Amazon Managed Streaming for Apache Kafka (Amazon MSK) unterstützt jetzt Apache Kafka Version 3.3.1 für neue und bestehende Cluster. Apache Kafka 3.3.1 enthält mehrere Bugfixes und neue Funktionen, die die Leistung verbessern. Zu den wichtigsten Funktionen gehören Verbesserungen an Metriken und Partitionierung. Amazon MSK wird in dieser Version aus Stabilitätsgründen weiterhin Zookeeper für die Quorumverwaltung verwenden und verwalten. Eine vollständige Liste der Verbesserungen und Bugfixes finden Sie in den Apache Kafka-Versionshinweisen für 3.3.1.

Weitere Informationen zu Apache Kafka Version 3.3.1 finden Sie in den Versionshinweisen auf der Download-Seite von Apache Kafka.

Amazon MSK versie 3.1.1

Amazon Managed Streaming for Apache Kafka (Amazon MSK) unterstützt jetzt Apache Kafka Version 3.1.1 und 3.2.0 für neue und bestehende Cluster. Apache Kafka 3.1.1 und Apache Kafka 3.2.0 enthalten mehrere Bugfixes und neue Funktionen, die die Leistung verbessern. Zu den wichtigsten Funktionen gehören Verbesserungen der Metriken und die Verwendung von Themen-IDs. MSK wird Zookeeper in dieser Version aus Stabilitätsgründen weiterhin für die Quorumverwaltung verwenden und verwalten. Eine vollständige Liste der Verbesserungen und Bugfixes finden Sie in den Apache Kafka-Versionshinweisen für 3.1.1 und 3.2.0.

Informationen zu Apache Kafka Version 3.1.1 und 3.2.0 finden Sie in den Versionshinweisen zu 3.2.0 und 3.1.1 auf der Apache Kafka-Downloadseite.

Mehrstufiger Speicher von Amazon MSK, Version 2.8.2.tiered

Bei dieser Version handelt es sich um eine reine Amazon-MSK-Version von Apache Kafka Version 2.8.2 und sie ist mit Open-Source-Apache-Kafka-Clients kompatibel.

Die Version 2.8.2.tiered enthält Funktionen für gestaffelte Speicherung, die mit den in KIP-405 für Apache Kafka eingeführten APIs kompatibel sind. Weitere Informationen zu den Feature für gestaffelten Speicher für Amazon MSK finden Sie unter Gestaffelte Speicherung.

Apache Kafka Version 2.5.1

Apache Kafka Version 2.5.1 enthält mehrere Bugfixes und neue Funktionen, darunter Verschlüsselung bei der Übertragung für Apache und Administrationsclients. ZooKeeper Amazon

MSK stellt ZooKeeper TLS-Endpunkte bereit, die Sie während des DescribeCluster Vorgangs abfragen können.

Die Ausgabe des DescribeClusterVorgangs umfasst den `ZookeeperConnectStringTls` Knoten, der die TLS-Zookeeper-Endpunkte auflistet.

Das folgende Beispiel zeigt den `ZookeeperConnectStringTls`-Knoten der Antwort für den `DescribeCluster`-Vorgang:

```
"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"
```

Informationen zum Verwenden von TLS-Verschlüsselung mit Zookeeper finden Sie unter Verwendung der TLS-Sicherheit mit Apache ZooKeeper.

Weitere Informationen zu Apache Kafka Version 2.5.1 finden Sie in den Versionshinweisen auf der Download-Seite von Apache Kafka.

Amazon-MSK-Bugfix Version 2.4.1.1

Bei dieser Version handelt es sich um eine reine Amazon-MSK-Bugfix-Version von Apache Kafka Version 2.4.1. Diese Bugfix-Version enthält eine Lösung für KAFKA-9752, ein seltenes Problem, das dazu führt, dass Verbrauchergruppen ständig das Gleichgewicht wiederherstellen und den Status `PreparingRebalance` beibehalten. Dieses Problem betrifft Cluster, auf denen die Versionen 2.3.1 und 2.4.1 von Apache Kafka ausgeführt werden. Diese Version enthält einen von der Community erstellten Fix, der in Apache Kafka Version 2.5.0 verfügbar ist.

> **Note**
>
> Amazon-MSK-Cluster, auf denen Version 2.4.1.1 ausgeführt wird, sind mit jedem Apache-Kafka-Client kompatibel, der mit Apache Kafka Version 2.4.1 kompatibel ist.

Wir empfehlen, die MSK-Bugfix-Version 2.4.1.1 für neue Amazon-MSK-Cluster zu verwenden, wenn Sie Apache Kafka 2.4.1 bevorzugen. Sie können bestehende Cluster, auf denen Apache Kafka Version 2.4.1 ausgeführt wird, auf diese Version aktualisieren, um diesen Fix zu integrieren. Hinweise

zum Aktualisieren eines vorhandenen Clusters finden Sie unter Aktualisieren der Apache Kafka-Version.

Informationen zur Umgehung dieses Problems, ohne den Cluster auf Version 2.4.1.1 zu aktualisieren, finden Sie im Abschnitt Verbrauchergruppe steckt im Status `PreparingRebalance` fest des Fehlerbehebung bei Ihrem Amazon-MSK-Cluster-Handbuchs.

Apache Kafka Version 2.4.1 (verwenden Sie stattdessen 2.4.1.1)

> **Note**
>
> Mit Apache Kafka Version 2.4.1 können Sie keinen MSK-Cluster mehr erstellen. Stattdessen können Sie Amazon-MSK-Bugfix Version 2.4.1.1 mit Clients verwenden, die mit Apache Kafka Version 2.4.1 kompatibel sind. Und wenn Sie bereits einen MSK-Cluster mit Apache Kafka Version 2.4.1 haben, empfehlen wir Ihnen, ihn so zu aktualisieren, dass er stattdessen Apache Kafka Version 2.4.1.1 verwendet.

KIP-392 ist einer der wichtigsten Verbesserungen für Kafka, die in der Version 2.4.1 von Apache Kafka enthalten sind. Sie ermöglicht Konsumenten das Abrufen vom nächstgelegenen Replikat. Um dieses Feature zu verwenden, legen Sie `client.rack` in den Konsumenteneigenschaften auf die ID der Availability Zone des Konsumenten fest. Ein Beispiel für eine AZ-ID ist `use1-az1`. Amazon MSK legt `broker.rack` auf die IDs der Availability Zones der Broker fest. Sie müssen auch die Konfigurationseigenschaft `replica.selector.class` auf `org.apache.kafka.common.replica.RackAwareReplicaSelector` festlegen. Dabei handelt es sich um eine Implementierung für Rackinformationen von Apache Kafka.

Wenn Sie diese Version von Apache Kafka verwenden, werden die Metriken in der Überwachungsebene `PER_TOPIC_PER_BROKER` erst angezeigt, nachdem ihre Werte zum ersten Mal ungleich Null sind. Weitere Informationen hierzu finden Sie unter the section called “Überwachung auf `PER_TOPIC_PER_BROKER`-Ebene”.

Informationen zum Auffinden von Availability Zone-IDs finden Sie im AWS Resource Access Manager Benutzerhandbuch unter AZ-IDs für Ihre Ressource.

Informationen zum Festlegen von Konfigurationseigenschaften finden Sie unter Konfiguration.

Weitere Informationen zu KIP-392 finden Sie auf den Confluence-Seiten unter Allow Consumers to Fetch from Closest Replica.

Weitere Informationen zu Apache Kafka Version 2.4.1 finden Sie in den Versionshinweisen auf der Download-Seite von Apache Kafka.

Unterstützung für Amazon MSK-Versionen

In diesem Thema werden die Support-Richtlinie für Amazon MSK-Versionen und das Verfahren für Aktualisieren der Apache Kafka-Version beschrieben. Wenn Sie Ihre Kafka-Version aktualisieren, befolgen Sie die unter beschriebenen bewährten Methoden. Bewährte Methoden für Versionsupgrades

Support-Richtlinie für Amazon MSK-Versionen

In diesem Abschnitt werden die Support-Richtlinien für von Amazon MSK unterstützte Kafka-Versionen beschrieben.

- Alle Kafka-Versionen werden bis zum Ende des Supports unterstützt. Einzelheiten zu den Terminen, an denen der Support endet, finden Sie unterUnterstützte Apache Kafka-Versionen. Führen Sie vor Ablauf des Supportzeitraums ein Upgrade Ihres MSK-Clusters auf die empfohlene Kafka-Version oder eine höhere Version durch. Einzelheiten zur Aktualisierung Ihrer Apache Kafka-Version finden Sie unter. Aktualisieren der Apache Kafka-Version Ein Cluster, der nach Ablauf des Supports eine Kafka-Version verwendet, wird automatisch auf die empfohlene Kafka-Version aktualisiert.
- MSK wird die Unterstützung für neu erstellte Cluster, die Kafka-Versionen mit veröffentlichten Enddaten für den Support verwenden, schrittweise einstellen.

Aktualisieren der Apache Kafka-Version

Sie können einen vorhandenen MSK-Cluster auf eine neuere Version von Apache Kafka aktualisieren. Sie können es nicht auf eine ältere Version aktualisieren. Wenn Sie die Apache-Kafka-Version eines MSK-Clusters aktualisieren, überprüfen Sie auch Ihre clientseitige Software, um sicherzustellen, dass Sie mit ihrer Version die Funktionen der neuen Apache-Kafka-Version des Clusters nutzen können. Amazon MSK aktualisiert nur die Serversoftware. Es aktualisiert Ihre Clients nicht.

Weitere Informationen zum Hochverfügbarmachen eines Clusters während eines Updates finden Sie unter the section called “Erstellen hochverfügbarer Cluster”.

> **Important**
>
> Sie können die Apache-Kafka-Version für einen MSK-Cluster nicht aktualisieren, der die in the section called “ Die Größe Ihres Clusters anpassen: Anzahl der Partitionen pro Broker” beschriebenen Grenzwerte überschreitet.

Aktualisierung der Apache Kafka-Version mit dem AWS Management Console

1. Öffnen Sie die Amazon-MSK-Konsole unter https://console.aws.amazon.com/msk/.
2. Wählen Sie den MSK-Cluster aus, auf dem Sie die Apache-Kafka-Version aktualisieren möchten.
3. Wählen Sie auf der Registerkarte Eigenschaften im Abschnitt Apache-Kafka-Version die Option Aktualisieren.

Aktualisierung der Apache Kafka-Version mit dem AWS CLI

1. Führen Sie den folgenden Befehl aus und *ClusterArn*ersetzen Sie ihn durch den Amazon-Ressourcennamen (ARN), den Sie bei der Erstellung Ihres Clusters erhalten haben. Wenn Ihnen der ARN für Ihren Cluster nicht vorliegt, finden Sie ihn, indem Sie alle Cluster auflisten. Weitere Informationen finden Sie unter the section called “Auflisten von Clustern”.

   ```
   aws kafka get-compatible-kafka-versions --cluster-arn ClusterArn
   ```

   Die Ausgabe dieses Befehls enthält eine Liste der Apache-Kafka-Versionen, auf die Sie den Cluster aktualisieren können. Es sollte wie das folgende Beispiel aussehen.

   ```
   {
       "CompatibleKafkaVersions": [
           {
               "SourceVersion": "2.2.1",
               "TargetVersions": [
                   "2.3.1",
                   "2.4.1",
                   "2.4.1.1",
                   "2.5.1"
               ]
           }
       ]
   ```

```
}
```

2. Führen Sie den folgenden Befehl aus und *ClusterArn*ersetzen Sie ihn durch den Amazon-Ressourcennamen (ARN), den Sie bei der Erstellung Ihres Clusters erhalten haben. Wenn Ihnen der ARN für Ihren Cluster nicht vorliegt, finden Sie ihn, indem Sie alle Cluster auflisten. Weitere Informationen finden Sie unter the section called “Auflisten von Clustern”.

   Ersetzen Sie *Aktuelle-Cluster-Version* durch die aktuelle Version des Clusters. Denn *TargetVersion*Sie können jede der Zielversionen aus der Ausgabe des vorherigen Befehls angeben.

   > **Important**
   >
   > Cluster-Versionen sind keine einfachen Ganzzahlen. Verwenden Sie den Befehl DescribeClusteroperation oder describe-cluster, um die aktuelle Version des Clusters AWS CLI zu finden. KTVPDKIKX0DER ist ein Beispiel für eine Version.

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

   Die Ausgabe des Befehls sieht wie das folgende JSON aus.

```
{

    "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. Um das Ergebnis des `update-cluster-kafka-version` Vorgangs zu erhalten, führen Sie den folgenden Befehl aus und ersetzen Sie *ClusterOperationArn* durch den ARN, den Sie in der Ausgabe des `update-cluster-kafka-version` Befehls erhalten haben.

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

Die Ausgabe dieses `describe-cluster-operation`-Befehls sieht wie das folgende JSON-Beispiel aus.

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

Wenn `OperationState` den Wert „`UPDATE_IN_PROGRESS`“ aufweist, warten Sie eine Weile, bevor Sie den `describe-cluster-operation`-Befehl erneut ausführen. Wenn der Vorgang abgeschlossen ist, erhält `OperationState` den Wert `UPDATE_COMPLETE`. Da die Zeit, die Amazon MSK benötigt, um den Vorgang abzuschließen, unterschiedlich ist, müssen Sie dies möglicherweise wiederholt überprüfen, bis der Vorgang abgeschlossen ist.

Aktualisieren der Apache Kafka Version mit der API

1. Rufen Sie den GetCompatibleKafkaVersionsVorgang auf, um eine Liste der Apache Kafka-Versionen abzurufen, auf die Sie den Cluster aktualisieren können.
2. Rufen Sie den UpdateClusterKafkaVersionVorgang auf, um den Cluster auf eine der kompatiblen Apache Kafka-Versionen zu aktualisieren.

Bewährte Methoden für Versionsupgrades

Um die Kontinuität der Clients während des fortlaufenden Updates sicherzustellen, das im Rahmen des Kafka-Versionsupgrade-Prozesses durchgeführt wird, sollten Sie die Konfiguration Ihrer Clients und die Themen zu Apache Kafka wie folgt überprüfen:

- Stellen Sie den Themenreplikationsfaktor (RF) auf einen Mindestwert von 2 für Zwei-AZ-Cluster und einen Mindestwert von 3 für Drei-AZ-Cluster ein. Ein RF-Wert von 2 kann dazu führen, dass Partitionen während des Patchens offline sind.
- Legen Sie die minimale Anzahl synchronisierter Replikate (minISR) auf einen Höchstwert von fest, `RF - 1` um sicherzustellen, dass der Partitionsreplikatsatz tolerieren kann, dass ein Replikat offline oder zu wenig repliziert ist.
- Konfigurieren Sie Clients so, dass sie mehrere Broker-Verbindungszeichenfolgen verwenden. Wenn die Verbindungszeichenfolge eines Clients mehrere Broker enthält, kann ein Failover durchgeführt werden, wenn ein bestimmter Broker, der Client-I/O unterstützt, gepatcht wird. Informationen zum Abrufen einer Verbindungszeichenfolge mit mehreren Brokern finden Sie unter Bootstrap-Broker für einen Amazon MSK-Cluster abrufen.
- Wir empfehlen, die Verbindungsclients auf die empfohlene Version oder höher zu aktualisieren, um von den Funktionen der neuen Version zu profitieren. Client-Upgrades unterliegen nicht dem Ende der Lebensdauer (EOL) der Kafka-Version Ihres MSK-Clusters und müssen auch nicht bis zum EOL-Datum abgeschlossen sein. Apache Kafka bietet eine bidirektionale Client-

Kompatibilitätsrichtlinie, die es älteren Clients ermöglicht, mit neueren Clustern zu arbeiten und umgekehrt.

- Kafka-Clients, die die Versionen 3.x.x verwenden, verfügen wahrscheinlich über die folgenden Standardwerte: und. `acks=all` `enable.idempotence=true` `acks=all`unterscheidet sich von der vorherigen Standardeinstellung von `acks=1` und bietet zusätzliche Haltbarkeit, indem sichergestellt wird, dass alle synchronisierten Replikate die Produktionsanforderung bestätigen. In ähnlicher Weise `enable.idempotence` war die Standardeinstellung für zuvor. `false` Die Änderung `enable.idempotence=true` zur Standardeinstellung verringert die Wahrscheinlichkeit doppelter Nachrichten. Diese Änderungen gelten als bewährte Einstellungen und können zu einer geringen zusätzlichen Latenz führen, die innerhalb der normalen Leistungsparameter liegt.
- Verwenden Sie die empfohlene Kafka-Version, wenn Sie neue MSK-Cluster erstellen. Wenn Sie die empfohlene Kafka-Version verwenden, können Sie von den neuesten Kafka- und MSK-Funktionen profitieren.

Fehlerbehebung bei Ihrem Amazon-MSK-Cluster

Die folgenden Informationen können zum Beheben von Problemen mit Ihrem Amazon-MSK-Cluster nützlich sein. Sie können Ihr Problem auch im AWS re:Post posten.

Themen

Der Austausch eines Volumes führt aufgrund einer Überlastung der Replikation zu einer Überlastung der Festplatte

Bei einem ungeplanten Ausfall der Volume-Hardware kann Amazon MSK das Volume durch eine neue Instance ersetzen. Kafka füllt das neue Volume erneut auf, indem es Partitionen von anderen Brokern im Cluster repliziert. Sobald Partitionen repliziert und aufgeholt wurden, kommen sie für eine Leadership- und ISR-Mitgliedschaft (In-Sync Replica) in Frage.

Problem

Bei einem Broker, der sich nach dem Austausch eines Volumes erholt, können einige Partitionen unterschiedlicher Größe vor anderen wieder online gehen. Dies kann problematisch sein, da diese Partitionen Datenverkehr von demselben Broker bereitstellen können, der immer noch andere Partitionen abholt (repliziert). Dieser Replikationsverkehr kann manchmal die zugrundeliegenden Volumendurchsatzgrenzen, die im Standardfall 250 MiB pro Sekunde betragen, sättigen. Wenn diese Sättigung eintritt, sind alle Partitionen betroffen, die bereits abgeholt wurden. Dies führt zu einer Latenz im gesamten Cluster bei allen Brokern, die ISR mit den aufgenommenen Partitionen teilen (nicht nur bei Leader-Partitionen aufgrund von Remote-Acks). `acks=all` Dieses Problem tritt häufiger bei größeren Clustern auf, die eine größere Anzahl von Partitionen mit unterschiedlicher Größe haben.

Empfehlung

- Um den I/O-Status der Replikation zu verbessern, stellen Sie sicher, dass die Thread-Einstellungen nach bewährten Methoden vorhanden sind.
- Um die Wahrscheinlichkeit einer zugrundeliegenden Volumensättigung zu verringern, sollten Sie bereitgestellten Speicher mit einem höheren Durchsatz aktivieren. Für Replikationsfälle mit hohem Durchsatz wird ein Mindestdurchsatz von 500 MiB/s empfohlen, der tatsächlich benötigte Wert hängt jedoch vom Durchsatz und vom Anwendungsfall ab. Bereitstellen des Speicherdurchsatzes.
- Um den Replikationsdruck `num.replica.fetchers` zu minimieren, senken Sie den Wert auf den Standardwert von2.

Verbrauchergruppe steckt im Status **`PreparingRebalance`** fest

Wenn sich eine oder mehrere Ihrer Verbrauchergruppen in einem Zustand der ständigen Neuausrichtung befinden, kann dies am Apache-Kafka-Problem KAFKA-9752 liegen, das die Apache-Kafka-Versionen 2.3.1 und 2.4.1 betrifft.

Um dieses Problem zu beheben, empfehlen wir Ihnen, Ihren Cluster auf die Version Amazon-MSK-Bugfix Version 2.4.1.1 zu aktualisieren, die eine Lösung für dieses Problem enthält. Informationen zur Aktualisierung eines vorhandenen Clusters auf die Amazon-MSK-Bugfix-Version 2.4.1.1 finden Sie unter Aktualisieren der Apache Kafka-Version.

Um dieses Problem zu lösen, ohne den Cluster auf die Bug-Fix-Version 2.4.1.1 des Amazon MSK zu aktualisieren, müssen Sie entweder die Kafka-Clients für die Verwendung von Static-Membership-Protokoll einrichten oder den koordinierenden Broker-Knoten der festgefahrenen Verbrauchergruppe auf Identifizieren und neu starten einstellen.

Implementierung des Static-Membership-Protokolls

Gehen Sie folgendermaßen vor, um das Static-Membership-Protokoll in Ihren Clients zu implementieren:

1. Setzen Sie die `group.instance.id`-Eigenschaft Ihrer Kafka-Verbraucher-Konfiguration auf eine statische Zeichenfolge, die den Verbraucher in der Gruppe identifiziert.
2. Stellen Sie sicher, dass andere Instances der Konfiguration aktualisiert werden, sodass sie die statische Zeichenfolge verwenden.
3. Stellen Sie die Änderungen für Ihre Kafka-Verbraucher bereit.

Die Verwendung des Static Membership Protocol ist effektiver, wenn das Sitzungs-Timeout in der Client-Konfiguration auf eine Dauer festgelegt ist, die es dem Verbraucher ermöglicht, sich zu erholen, ohne vorzeitig eine Neuverteilung der Verbrauchergruppen auszulösen. Wenn Ihre Verbraucheranwendung beispielsweise eine Nichtverfügbarkeit von 5 Minuten toleriert, wäre ein angemessener Wert für das Sitzungs-Timeout 4 Minuten anstelle des Standardwerts von 10 Sekunden.

> **Note**
>
> Die Verwendung des Static-Membership-Protokolls verringert nur die Wahrscheinlichkeit, dass dieses Problem auftritt. Dieses Problem kann auch dann auftreten, wenn Sie das Static-Membership-Protokoll verwenden.

Den koordinierenden Broker-Knoten neu starten

Gehen Sie wie folgt vor, um den koordinierenden Broker-Knoten neu zu starten:

1. Identifizieren Sie den Gruppenkoordinator mithilfe des Befehls `kafka-consumer-groups.sh`.
2. Starten Sie den Gruppenkoordinator der festgefahrenen Nutzergruppe mithilfe der RebootBrokerAPI-Aktion neu.

Fehler beim Senden von Brokerprotokollen an Amazon CloudWatch Logs

Wenn Sie versuchen, Ihren Cluster so einzurichten, dass er Broker-Logs an Amazon CloudWatch Logs sendet, kann es zu einer von zwei Ausnahmen kommen.

Wenn Sie die Ausnahme `InvalidInput.LengthOfCloudWatchResourcePolicyLimitExceeded` erhalten, wiederholen Sie den Vorgang, verwenden jedoch Protokollgruppen, die mit `/aws/vendedlogs/` beginnen. Weitere Informationen finden Sie unter Aktivieren der Protokollierung aus bestimmten Amazon Web Services.

Wenn Sie eine `InvalidInput.NumberOfCloudWatchResourcePoliciesLimitExceeded` Ausnahme erhalten, wählen Sie eine bestehende Amazon CloudWatch Logs-Richtlinie in Ihrem Konto aus und hängen Sie die folgende JSON-Datei an.

```
{"Sid":"AWSLogDeliveryWrite","Effect":"Allow","Principal":
{"Service":"delivery.logs.amazonaws.com"},"Action":
["logs:CreateLogStream","logs:PutLogEvents"],"Resource":["*"]}
```

Wenn Sie versuchen, den obigen JSON-Code an eine bestehende Richtlinie anzuhängen, aber eine Fehlermeldung erhalten, die besagt, dass Sie die maximale Länge für die von Ihnen gewählte Richtlinie erreicht haben, versuchen Sie, den JSON an eine andere Ihrer Amazon CloudWatch Logs-Richtlinien anzuhängen. Nachdem Sie das JSON an eine bestehende Richtlinie angehängt haben, versuchen Sie erneut, die Broker-Log-Übermittlung an Amazon Logs einzurichten. CloudWatch

Keine Standard-Sicherheitsgruppe

Wenn Sie versuchen, einen Cluster zu erstellen und einen Fehler über das Fehlen einer Standardsicherheitsgruppe erhalten, verwenden Sie möglicherweise eine VPC, die für Sie freigegeben wurde. Bitten Sie Ihren Administrator, Ihnen die Berechtigung zur Beschreibung der Sicherheitsgruppen auf dieser VPC zu erteilen, und versuchen Sie es erneut. Ein Beispiel für eine

Richtlinie, die diese Aktion zulässt, finden Sie unter Amazon EC2: Ermöglicht es, die mit einer bestimmten VPC verknüpften EC2-Sicherheitsgruppen programmgesteuert und in der Konsole zu verwalten .

Der Cluster steckt anscheinend im Status „CREATING“ fest.

Gelegentlich dauert die Cluster-Erstellung bis zu 30 Minuten. Warten Sie 30 Minuten und überprüfen Sie den Status des Clusters erneut.

Der Cluster-Status wird von „CREATING“ in „FAILED“ geändert.

Versuchen Sie erneut, den Cluster zu erstellen.

Der Cluster-Status ist „ACTIVE“, Produzenten können jedoch keine Daten senden oder Konsumenten können keine Daten empfangen.

- Wenn die Cluster-Erstellung erfolgreich ist (der Cluster-Status lautet „`ACTIVE`“), Sie jedoch keine Daten senden oder empfangen können, stellen Sie sicher, dass Ihre Produzenten- und Konsumentenanwendungen auf den Cluster zugreifen können. Weitere Informationen finden Sie in der Anleitung in the section called “Schritt 3: Einen Client-Computer erstellen”.

- Wenn Ihre Produzenten und Konsumenten auf den Cluster zugreifen können, aber immer noch Probleme beim Erstellen und Nutzen von Daten auftreten, könnte dies durch KAFKA-7697 verursacht werden. Dies wirkt sich auf Apache Kafka Version 2.1.0 aus und kann zu einem Deadlock in einem oder mehreren Brokern führen. Ziehen Sie eine Migration zu Apache Kafka 2.2.1 in Betracht. Diese Version ist von diesem Fehler nicht betroffen. Weitere Informationen zur Migration finden Sie unter Migration.

AWS CLI erkennt Amazon MSK nicht

Wenn Sie das AWS CLI installiert haben, es aber die Amazon MSK-Befehle nicht erkennt, führen Sie ein Upgrade AWS CLI auf die neueste Version durch. Detaillierte Anweisungen zum Upgrade von finden Sie AWS CLI unter Installation von. AWS Command Line Interface Informationen zur Verwendung der Befehle AWS CLI zum Ausführen von Amazon MSK-Befehlen finden Sie unterFunktionsweise.

Partitionen werden auf „offline“ festgelegt oder Replikate sind nicht synchronisiert.

Dies können Anzeichen von wenig Speicherplatz sein. Siehe the section called “Wenig Speicherplatz”.

Wenig Speicherplatz

Lesen Sie die folgenden bewährten Methoden für die Verwaltung des Speicherplatzes: the section called “Überwachen der Festplattenkapazität” und the section called “Anpassen der Datenaufbewahrungsparameter”.

Wenig Arbeitsspeicher

Wenn Sie sehen, dass die `MemoryUsed`-Metrik hoch oder `MemoryFree` niedrig ist, deutet das nicht auf ein Problem hin. Apache Kafka wurde entwickelt, um so viel Speicher wie möglich zu verwenden, und es verwaltet ihn optimal.

Der Produzent erhält NotLeaderForPartitionException

Dies ist oft ein vorübergehender Fehler. Legen Sie den `retries`-Konfigurationsparameter des Produzenten auf einen Wert fest, der höher als sein aktueller Wert ist.

Die unterreplizierten Partitionen (URP) sind größer als Null

Die Überwachung der `UnderReplicatedPartitions`-Metrik ist wichtig. In einem fehlerfreien MSK-Cluster weist diese Metrik den Wert „0“ auf. Der Wert kann aus Folgenden Gründen größer als Null sein.

- Falls es zu Spitzenwerten bei `UnderReplicatedPartitions` kommt, wird der Cluster möglicherweise nicht in der richtigen Größe für die Verarbeitung von eingehendem und ausgehendem Datenverkehr bereitgestellt. Siehe Bewährte Methoden.
- Wenn `UnderReplicatedPartitions` auch in Zeiträumen mit geringem Datenverkehr konstant größer als 0 ist, haben Sie möglicherweise restriktive ACLs festgelegt, die Brokern keinen Themenzugriff gewähren. Zum Replizieren von Partitionen müssen Broker für die Themen „READ“

und „DESCRIBE“ autorisiert sein. „DESCRIBE“ wird standardmäßig mit der „READ“-Autorisierung erteilt. Informationen zum Festlegen von ACLs finden Sie unter Autorisierung und ACLs in der Apache Kafka-Dokumentation.

Der Cluster hat die Themen __amazon_msk_canary und __amazon_msk_canary_state

Möglicherweise sehen Sie, dass Ihr MSK-Cluster ein Thema mit dem Namen `__amazon_msk_canary` und ein anderes mit dem Namen `__amazon_msk_canary_state` hat. Dies sind interne Themen, die Amazon MSK erstellt und für Metriken zum Cluster-Zustand und zur Diagnose verwendet. Diese Themen haben eine vernachlässigbare Größe und können nicht gelöscht werden.

Die Partitionsreplikation schlägt fehl

Stellen Sie sicher, dass Sie keine ACLs für CLUSTER_ACTIONS festgelegt haben.

Es kann nicht auf einen Cluster zugegriffen werden, für den der öffentliche Zugriff aktiviert ist

Wenn für Ihren Cluster der öffentliche Zugriff aktiviert ist, Sie aber immer noch nicht über das Internet darauf zugreifen können, gehen Sie wie folgt vor:

1. Stellen Sie sicher, dass die Regeln der Sicherheitsgruppe für eingehenden Datenverkehr Ihre IP-Adresse und den Port des Clusters erlauben. Eine Liste der Cluster-Portnummern finden Sie unter the section called “Port-Informationen”. Stellen Sie außerdem sicher, dass die Regeln für ausgehenden Datenverkehr der Sicherheitsgruppe ausgehende Kommunikation zulassen. Weitere Informationen zu Sicherheitsgruppen und Regeln für eingehenden und ausgehenden Datenverkehr finden Sie unter Sicherheitsgruppen für Ihre VPC im Amazon-VPC-Benutzerhandbuch.
2. Stellen Sie sicher, dass Ihre IP-Adresse und der Port des Clusters in den Regeln für eingehenden Datenverkehr der VPC-Netzwerk-ACL des Clusters zulässig sind. Im Gegensatz zu Sicherheitsgruppen sind Netzwerk-ACLs zustandslos. Dies bedeutet, dass Sie die Regeln für den ein- und ausgehenden Datenverkehr konfigurieren müssen. Erlauben Sie in den Regeln für ausgehenden Datenverkehr den gesamten Datenverkehr (Portbereich: 0–65535) zu Ihrer

IP-Adresse zu. Weitere Informationen finden Sie unter Hinzufügen und Löschen von Regeln im Amazon-VPC-Benutzerhandbuch.

3. Stellen Sie sicher, dass Sie die Bootstrap-Brokers-Zeichenfolge mit öffentlichem Zugriff für den Zugriff auf den Cluster verwenden. Ein MSK-Cluster, für den der öffentliche Zugriff aktiviert ist, hat zwei verschiedene Bootstrap-Broker-Zeichenfolgen, eine für den öffentlichen Zugriff und eine für den Zugriff innerhalb AWS. Weitere Informationen finden Sie unter the section called “Holen Sie sich die Bootstrap-Broker mit dem AWS Management Console”.

Von innen kann nicht auf den Cluster zugegriffen werden AWS: Netzwerkprobleme

Wenn Sie über eine Apache-Kafka-Anwendung verfügen, die nicht erfolgreich mit einem MSK-Cluster kommunizieren kann, führen Sie zunächst den folgenden Konnektivitätstest durch.

1. Verwenden Sie eine der in the section called “Abrufen der Bootstrap-Broker” beschriebenen Methoden, um die Adressen der Bootstrap-Broker zu erhalten.
2. Im folgenden Befehl ersetzen Sie *bootstrap-broker* durch eine der Broker-Adressen, die Sie im vorherigen Schritt erhalten haben. Ersetzen Sie die *port-number* durch 9094, wenn der Cluster für die Verwendung der TLS-Authentifizierung eingerichtet ist. Wenn der Cluster keine TLS-Authentifizierung verwendet, ersetzen Sie *port-number* durch 9092. Führen Sie den Befehl vom Clientcomputer aus.

   ```
   telnet bootstrap-broker port-number
   ```

   Wobei die Portnummer wie folgt lautet:

   - 9094, wenn der Cluster für die Verwendung der TLS-Authentifizierung eingerichtet ist.
   - 9092 Wenn der Cluster keine TLS-Authentifizierung verwendet.
   - Eine andere Portnummer ist erforderlich, wenn der öffentliche Zugriff aktiviert ist.

   Führen Sie den Befehl vom Clientcomputer aus.
3. Wiederholen Sie den vorherigen Befehl für alle Bootstrap-Broker.

Wenn der Client-Computer auf die Broker zugreifen kann, bedeutet dies, dass keine Verbindungsprobleme vorliegen. Führen Sie in diesem Fall den folgenden Befehl aus, um zu

überprüfen, ob Ihr Apache Kafka Client korrekt eingerichtet ist. Um *bootstrap-brokers (Bootstrap-Broker)* zu erhalten, verwenden Sie eine der in the section called “Abrufen der Bootstrap-Broker” beschriebenen Methoden. Ersetzen Sie *topic* durch den Namen Ihres Themas.

```
<path-to-your-kafka-installation>/bin/kafka-console-producer.sh --broker-
list bootstrap-brokers --producer.config client.properties --topic topic
```

Wenn der vorherige Befehl erfolgreich ist, bedeutet dies, dass Ihr Client korrekt eingerichtet ist. Wenn Sie immer noch nicht in der Lage sind, aus einer Anwendung zu produzieren und zu konsumieren, debuggen Sie das Problem auf Anwendungsebene.

Wenn der Client-Computer nicht auf die Broker zugreifen kann, finden Sie in den folgenden Unterabschnitten Anleitungen, die auf Ihrem Client-Computer-Setup basieren.

Amazon-EC2-Client und MSK-Cluster in derselben VPC

Wenn sich der Client-Computer in derselben VPC wie der MSK-Cluster befindet, stellen Sie sicher, dass die Sicherheitsgruppe des Clusters über eine Regel für eingehenden Datenverkehr verfügt, die Datenverkehr von der Sicherheitsgruppe des Client-Computers akzeptiert. Informationen zum Einrichten dieser Regeln finden Sie unter Sicherheitsgruppenregeln. Ein Beispiel für den Zugriff auf einen Cluster von einer Amazon-EC2-Instance aus, die sich in derselben VPC wie der Cluster befindet, finden Sie unter Erste Schritte.

Amazon-EC2-Client und MSK-Cluster in verschiedenen VPCs

Wenn sich der Clientcomputer und der Cluster in zwei verschiedenen VPCs befinden, stellen Sie Folgendes sicher:

- Die beiden VPCs sind durch Peering verbunden.
- Der Status der Peering-Verbindung ist aktiv.
- Die Routingtabellen der beiden VPCs sind korrekt eingerichtet.

Weitere Informationen zum VPC-Peering finden Sie unter Arbeiten mit VPC-Peering-Verbindungen.

On-Premises-Client

Stellen Sie bei einem lokalen Client, der so eingerichtet ist, dass er eine Verbindung zum MSK-Cluster herstellt, Folgendes sicher: AWS VPN

- Der VPN-Verbindungsstatus lautet UP. Informationen zum Überprüfen des VPN-Verbindungsstatus finden Sie unter Wie überprüfe ich den aktuellen Status meines VPN-Tunnels?.
- Die Routingtabelle der VPC des Clusters enthält die Route für einen On-Premises-CIDR, dessen Ziel das Format `Virtual private gateway(vgw-xxxxxxxx)` aufweist.
- Die Sicherheitsgruppe des MSK-Clusters erlaubt Datenverkehr auf Port 2181, Port 9092 (wenn Ihr Cluster Nur-Text-Datenverkehr akzeptiert) und Port 9094 (wenn Ihr Cluster TLS-verschlüsselten Datenverkehr akzeptiert).

Weitere Anleitungen AWS VPN zur Fehlerbehebung finden Sie unter Fehlerbehebung bei Client VPN.

AWS Direct Connect

Wenn der Client verwendet AWS Direct Connect, finden Sie weitere Informationen unter Problembehandlung AWS Direct Connect.

Wenn die vorherige Anleitung zur Fehlerbehebung das Problem nicht beheben kann, stellen Sie sicher, dass keine Firewall den Netzwerkverkehr blockiert. Verwenden Sie zum weiteren Debuggen Tools wie `tcpdump` und `Wireshark` zum Analysieren des Datenverkehrs und stellen Sie sicher, dass er den MSK-Cluster erreicht.

Fehlgeschlagene Authentifizierung: Zu viele Verbindungen

Der Fehler `Failed authentication ... Too many connects` weist darauf hin, dass ein Broker sich selbst schützt, weil ein oder mehrere IAM-Clients mit einer aggressiv-schnellen Rate versuchen, eine Verbindung zu ihm herzustellen. Um Brokern zu helfen, eine höhere Rate neuer IAM-Verbindungen zu akzeptieren, können Sie den Konfigurationsparameter `reconnect.backoff.ms` erhöhen.

Weitere Informationen zu den Ratenlimits für neue Verbindungen pro Broker finden Sie auf der Amazon-MSK-Kontingent-Seite.

MSK Serverless: Die Cluster-Erstellung schlägt fehl

Wenn Sie versuchen, einen MSK-Serverless-Cluster zu erstellen, und der Workflow fehlschlägt, sind Sie möglicherweise nicht berechtigt, einen VPC-Endpunkt zu erstellen. Stellen Sie sicher, dass Ihr Administrator Ihnen die Berechtigung erteilt hat, einen VPC-Endpunkt zu erstellen, indem Sie die `ec2:CreateVpcEndpoint`-Aktion zulassen.

Eine vollständige Liste der Berechtigungen, die für die Ausführung aller Amazon-MSK-Aktionen erforderlich sind, finden Sie unter AWS verwaltete Richtlinie: AmazonMSK FullAccess.

Bewährte Methoden

In diesem Thema werden einige bewährte Methoden beschrieben, die bei der Verwendung von Amazon MSK zu beachten sind.

Die Größe Ihres Clusters anpassen: Anzahl der Partitionen pro Broker

Die folgende Tabelle zeigt die empfohlene maximale Anzahl von Partitionen (einschließlich Leader- und Follower-Replikate) pro Broker.

Größe des Maklers	Empfohlene maximale Anzahl von Partitionen (einschließlich Leader- und Follower-Replikate) pro Broker
`kafka.t3.small`	300
`kafka.m5.large` oder `kafka.m5.xlarge`	1000
`kafka.m5.2xlarge`	2000
`kafka.m5.4xlarge` , `kafka.m5.8xlarge` , `kafka.m5.12xlarge` , `kafka.m5.16xlarge` oder `kafka.m5.24xlarge`	4000
`kafka.m7g.large` oder `kafka.m7g.xlarge`	1000
`kafka.m7g.2xlarge`	2000
`kafka.m7g.4xlarge` , `kafka.m7g.8xlarge` `kafka.m7g.12xlarge` , oder `kafka.m7g.16xlarge`	4000

Wenn die Anzahl der Partitionen pro Broker den empfohlenen Wert überschreitet und Ihr Cluster überlastet ist, können Sie möglicherweise die folgenden Vorgänge nicht ausführen:

- Die Cluster-Konfiguration aktualisieren
- Aktualisieren Sie den Cluster auf eine kleinere Broker-Größe
- Ordnen Sie einem Cluster mit SASL/SCRAM-Authentifizierung ein AWS Secrets Manager Geheimnis zu

Eine hohe Anzahl von Partitionen kann auch dazu führen, dass Kafka-Metriken beim CloudWatch und beim Prometheus-Scraping fehlen.

Eine Anleitung zur Auswahl der Anzahl der Partitionen finden Sie unter Apache Kafka unterstützt 200K Partitionen pro Cluster. Wir empfehlen Ihnen außerdem, Ihre eigenen Tests durchzuführen, um die richtige Größe für Ihre Broker zu ermitteln. Weitere Informationen zu den verschiedenen Brokergrößen finden Sie unterthe section called “Größen der Makler”.

Die Größe Ihres Clusters anpassen: Anzahl der Broker pro Cluster

Informationen zur Ermittlung der richtigen Anzahl von Brokern für Ihren MSK-Cluster und zum Verständnis der Kosten finden Sie in der Tabelle MSK: Dimensionierung und Preise. Diese Tabelle enthält eine Schätzung für die Dimensionierung eines MSK-Clusters und die damit verbundenen Kosten von Amazon MSK im Vergleich zu einem ähnlichen, selbstverwalteten, EC2-basierten Apache-Kafka-Cluster. Weitere Informationen zu den Eingabeparametern in der Tabelle erhalten Sie, wenn Sie den Mauszeiger über die Parameterbeschreibungen bewegen. Die Schätzungen in dieser Tabelle sind konservativ und bieten einen Ausgangspunkt für einen neuen Cluster. Leistung, Größe und Kosten des Clusters hängen von Ihrem Anwendungsfall ab. Wir empfehlen Ihnen, diese Werte anhand von Tests zu überprüfen.

Informationen darüber, wie sich die zugrunde liegende Infrastruktur auf die Leistung von Apache Kafka auswirkt, finden Sie im AWS Big Data-Blog unter Bewährte Methoden für die richtige Dimensionierung Ihrer Apache Kafka-Cluster zur Optimierung von Leistung und Kosten. Der Blogbeitrag enthält Informationen darüber, wie Sie Ihre Cluster so dimensionieren können, dass sie Ihren Durchsatz-, Verfügbarkeits- und Latenzanforderungen entsprechen. Der Beitrag enthält auch Antworten auf Fragen wie wann Sie hochskalieren im Vergleich zu aufskalieren sollten, sowie Anleitungen, wie Sie die Größe Ihrer Produktions-Cluster kontinuierlich überprüfen können.

Optimieren Sie den Cluster-Durchsatz für m5.4xl-, m7g.4xl- oder größere Instances

Wenn Sie m5.4xl-, m7g.4xl- oder größere Instances verwenden, können Sie den Cluster-Durchsatz optimieren, indem Sie die Konfigurationen num.io.threads und num.network.threads optimieren.

num.io.threads ist die Anzahl der Threads, die ein Broker für die Verarbeitung von Anfragen verwendet. Durch das Hinzufügen weiterer Threads bis zur Anzahl der für die Instanzgröße unterstützten CPU-Kerne kann der Cluster-Durchsatz verbessert werden.

num.network.threads ist die Anzahl der Threads, die der Broker für den Empfang aller eingehenden Anfragen und die Rückgabe von Antworten verwendet. Netzwerk-Threads platzieren eingehende Anfragen in einer Anforderungswarteschlange zur Verarbeitung durch io.threads. Wenn num.network.threads auf die Hälfte der Anzahl der für die Instanzgröße unterstützten CPU-Kerne festgelegt wird, kann die neue Instanzgröße voll genutzt werden.

> **Important**
>
> Erhöhen Sie num.network.threads nicht, ohne zuerst num.io.threads zu erhöhen, da dies zu einer Überlastung der Warteschlange führen kann.

Empfohlene Einstellungen

Instance-Größe	Empfohlener Wert für num.io.threads	Empfohlener Wert für 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

Instance-Größe	Empfohlener Wert für num.io.threads	Empfohlener Wert für num.network.threads
m7g.12xlarge	48	24
m7g.16xlarge	64	32

Verwenden Sie die neueste Version von Kafka, um Probleme mit nicht übereinstimmenden AdminClient Themen-IDs zu vermeiden

Die ID eines Themas geht verloren (Fehler: stimmt nicht mit der Themen-ID für die Partition überein), wenn Sie eine AdminClient Kafka-Version unter 2.8.0 mit dem Flag `--zookeeper` zum Erhöhen oder Neuzuweisen von Themenpartitionen für einen Cluster verwenden, der Kafka-Version 2.8.0 oder höher verwendet. Beachten Sie, dass das Flag `--zookeeper` in Kafka 2.5 veraltet ist und ab Kafka 3.0 entfernt wird. Siehe Aktualisieren von einer beliebigen Version 0.8.x bis 2.4.x auf 2.5.0.

Um eine Nichtübereinstimmung der Themen-IDs zu vermeiden, verwenden Sie einen Kafka-Client der Version 2.8.0 oder höher für Kafka-Admin-Vorgänge. Alternativ können Clients 2.5 und höher das Flag `--bootstrap-servers` anstelle des Flags `--zookeeper` verwenden.

Erstellen hochverfügbarer Cluster

Verwenden Sie die folgenden Empfehlungen, damit Ihr MSK-Cluster während eines Updates (z. B. wenn Sie die Broker-Größe oder die Apache Kafka-Version aktualisieren) oder wenn Amazon MSK einen Broker ersetzt, hochverfügbar ist.

- Richten Sie einen Drei-AZ-Cluster ein.
- Stellen Sie sicher, dass der Replikationsfaktor (RF) mindestens 3 beträgt. Beachten Sie, dass ein RF von 1 während eines fortlaufenden Updates zu Offline-Partitionen führen kann und ein RF von 2 zu Datenverlust führen kann.
- Legen Sie minimale In-Sync-Replikate (minISR) auf höchstens RF - 1 fest. Ein minISR, das dem RF entspricht, kann verhindern, dass das Erzeugen im Cluster während einer fortlaufenden Aktualisierung erfolgt. Mit einem minISR von 2 können dreiseitige replizierte Themen verfügbar sein, wenn ein Replikat offline ist.
- Stellen Sie sicher, dass die Client-Verbindungszeichenfolgen mindestens einen Broker aus jeder Availability Zone enthalten. Die Verwendung mehrerer Broker in der Verbindungszeichenfolge

eines Clients ermöglicht ein Failover, wenn ein bestimmter Broker für ein Update offline ist. Weitere Informationen zum Abrufen einer Verbindungszeichenfolge mit mehreren Brokern finden Sie unter the section called “Abrufen der Bootstrap-Broker”.

CPU-Auslastung überwachen

Amazon MSK empfiehlt dringend, die gesamte CPU-Auslastung für Ihre Broker (definiert als `CPU User + CPU System`) unter 60 % zu halten. Wenn mindestens 40 % der gesamten CPU Ihres Clusters verfügbar sind, kann Apache Kafka die CPU-Last bei Bedarf auf die Broker im Cluster verteilen. Ein Beispiel dafür, wann dies erforderlich ist, ist, wenn Amazon MSK einen Broker-Fehler erkennt und diesen behebt. In diesem Fall führt Amazon MSK automatische Wartungsarbeiten wie Patches durch. Ein anderes Beispiel ist, wenn ein Benutzer eine Änderung der Brokergröße oder ein Versionsupgrade anfordert. In diesen beiden Fällen stellt Amazon MSK fortlaufende Workflows bereit, die jeweils einen Broker offline schalten. Wenn Broker mit Lead-Partitionen offline gehen, weist Apache Kafka die Partitionsleitung neu zu, um die Arbeit auf andere Broker im Cluster umzuverteilen. Wenn Sie sich an diese bewährte Methode halten, können Sie sicherstellen, dass Ihr Cluster über genügend CPU-Reserven verfügt, um Betriebsereignisse wie diese zu tolerieren.

Sie können Amazon CloudWatch Metric Math verwenden, um eine zusammengesetzte Metrik zu erstellen, die `CPU User + CPU System` Stellen Sie einen Alarm ein, der ausgelöst wird, wenn die zusammengesetzte Metrik eine durchschnittliche CPU-Auslastung von 60 % erreicht. Wenn dieser Alarm ausgelöst wird, skalieren Sie den Cluster mit einer der folgenden Optionen:

- Option 1 (empfohlen): Aktualisieren Sie Ihre Broker-Größe auf die nächstgrößere Größe. Wenn die aktuelle Größe beispielsweise lautet`kafka.m5.large`, aktualisieren Sie den zu verwendenden Cluster`kafka.m5.xlarge`. Denken Sie daran, dass Amazon MSK, wenn Sie die Broker-Größe im Cluster aktualisieren, die Broker fortlaufend offline nimmt und vorübergehend die Partitionsführung anderen Brokern zuweist. Eine Größenaktualisierung dauert in der Regel 10–15 Minuten pro Broker.
- Option 2: Wenn es Themen gibt, in denen alle Nachrichten von Produzenten aufgenommen wurden, die Round-Robin-Schreibvorgänge verwenden (mit anderen Worten, Nachrichten sind nicht verschlüsselt und die Reihenfolge ist für Verbraucher nicht wichtig), erweitern Sie Ihren Cluster, indem Sie Broker hinzufügen. Fügen Sie außerdem Partitionen zu vorhandenen Themen mit dem höchsten Durchsatz hinzu. Verwenden Sie als Nächstes `kafka-topics.sh --describe`, um sicherzustellen, dass neu hinzugefügte Partitionen den neuen Brokern zugewiesen werden. Der Hauptvorteil dieser Option im Vergleich zur vorherigen Option besteht darin, dass Sie Ressourcen und Kosten detaillierter verwalten können. Darüber hinaus können Sie diese Option

verwenden, wenn die CPU-Auslastung deutlich über 60 % liegt, da diese Form der Skalierung in der Regel nicht zu einer erhöhten Belastung vorhandener Broker führt.

- Option 3: Erweitern Sie Ihren Cluster, indem Sie Broker hinzufügen, und weisen Sie dann vorhandene Partitionen neu zu mithilfe des Tools zur Neuzuweisung von Partitionen `kafka-reassign-partitions.sh`. Wenn Sie diese Option verwenden, muss der Cluster jedoch Ressourcen aufwenden, um Daten von Broker zu Broker zu replizieren, nachdem Partitionen neu zugewiesen wurden. Im Vergleich zu den beiden vorherigen Optionen kann dies die Belastung des Clusters zunächst erheblich erhöhen. Aus diesem Grund empfiehlt Amazon MSK, diese Option nicht zu verwenden, wenn die CPU-Auslastung über 70 % liegt, da die Replikation zu zusätzlicher CPU-Last und Netzwerk-Datenverkehr führt. Amazon MSK empfiehlt, diese Option nur zu verwenden, wenn die beiden vorherigen Optionen nicht durchführbar sind.

Weitere Empfehlungen:

- Überwachen Sie die gesamte CPU-Auslastung pro Broker als Proxy für die Lastverteilung. Wenn Broker eine durchweg ungleichmäßige CPU-Auslastung aufweisen, kann dies ein Zeichen dafür sein, dass die Last innerhalb des Clusters nicht gleichmäßig verteilt ist. Amazon MSK empfiehlt die Verwendung von Cruise Control, um die Lastverteilung über Partitionszuweisung kontinuierlich zu verwalten.
- Überwachen Sie die Latenz bei Produktion und Verbrauch. Die Latenz bei Produktion und Verbrauch kann linear mit der CPU-Auslastung zunehmen.
- JMX-Scrape-Intervall: Wenn Sie die offene Überwachung mit der Prometheus-Feature aktivieren, wird empfohlen, für Ihre Prometheus-Host-Konfiguration (prometheus.yml) ein Scrape-Intervall von 60 Sekunden oder höher (scrape_interval: 60s) zu verwenden. Eine Verkürzung des Scrape-Intervalls kann zu einer hohen CPU-Auslastung in Ihrem Cluster führen.

Überwachen der Festplattenkapazität

Um zu verhindern, dass der Speicherplatz für Nachrichten knapp wird, sollten Sie einen CloudWatch Alarm einrichten, der die Metrik überwacht. `KafkaDataLogsDiskUsed` Wenn der Wert dieser Metrik 85 % erreicht oder überschreitet, führen Sie eine oder mehrere der folgenden Aktionen aus:

- Verwenden Sie the section called “Auto Scaling”. Sie können den Broker-Speicher auch manuell erhöhen, wie unter the section called “Manuelle Skalierung” beschrieben.

- Verringern Sie den Aufbewahrungszeitraum für Nachrichten oder die Protokollgröße. Weitere Informationen hierzu finden Sie unter the section called “Anpassen der Datenaufbewahrungsparameter”.
- Löschen Sie nicht verwendete Themen.

Informationen zur Einrichtung und Verwendung von Alarmen finden Sie unter Amazon CloudWatch Alarms verwenden. Eine vollständige Liste der Amazon-MSK-Metriken finden Sie unter Überwachung eines Clusters.

Anpassen der Datenaufbewahrungsparameter

Durch die Verwendung von Nachrichten werden diese nicht aus dem Protokoll entfernt. Um regelmäßig Speicherplatz freizugeben, können Sie explizit einen Aufbewahrungszeitraum angeben, d. h., wie lange Nachrichten im Protokoll verbleiben. Sie können auch eine Größe für das Aufbewahrungsprotokoll angeben. Wenn entweder der Aufbewahrungszeitraum oder die Größe des Aufbewahrungsprotokolls erreicht ist, beginnt Apache Kafka, inaktive Segmente aus dem Protokoll zu entfernen.

Zum Angeben einer Aufbewahrungsrichtlinie auf Clusterebene legen Sie einen oder mehrere der folgenden Parameter fest: `log.retention.hours`, `log.retention.minutes`, `log.retention.ms` oder `log.retention.bytes`. Weitere Informationen finden Sie unter the section called “Benutzerdefinierte -Konfigurationen”.

Sie können Aufbewahrungsparameter auch auf Themenebene angeben:

- Verwenden Sie den folgenden Befehl, um einen Aufbewahrungszeitraum pro Thema anzugeben.

```
kafka-configs.sh --bootstrap-server $bs --alter --entity-type topics --entity-
name TopicName --add-config retention.ms=DesiredRetentionTimePeriod
```

- Verwenden Sie den folgenden Befehl, um eine Aufbewahrungsprotokollgröße pro Thema anzugeben.

```
kafka-configs.sh --bootstrap-server $bs --alter --entity-type topics --entity-
name TopicName --add-config retention.bytes=DesiredRetentionLogSize
```

Die auf Themenebene angegebenen Aufbewahrungsparameter haben Vorrang vor Parametern auf Clusterebene.

Beschleunigung der Protokollwiederherstellung nach einem unsauberen Herunterfahren

Nach einem unsauberen Herunterfahren kann es eine Weile dauern, bis ein Broker neu gestartet wird, da er die Protokollwiederherstellung durchführt. Standardmäßig verwendet Kafka nur einen einzigen Thread pro Protokollverzeichnis, um diese Wiederherstellung durchzuführen. Wenn Sie beispielsweise Tausende von Partitionen haben, kann die Protokollwiederherstellung Stunden dauern. Um die Protokollwiederherstellung zu beschleunigen, wird empfohlen, die Anzahl der Threads mithilfe der Konfigurationseigenschaft `num.recovery.threads.per.data.dir` zu erhöhen. Sie können es auf die Anzahl der CPU-Kerne einstellen.

Apache-Kafka-Arbeitsspeicher überwachen

Wir empfehlen, dass Sie den Arbeitsspeicher überwachen, den Apache Kafka verwendet. Andernfalls ist der Cluster möglicherweise nicht mehr verfügbar.

Um festzustellen, wie viel Arbeitsspeicher Apache Kafka verwendet, können Sie die `HeapMemoryAfterGC`-Metrik überwachen. `HeapMemoryAfterGC` ist der Prozentsatz des gesamten Heap-Speichers, der nach der Garbage Collection verwendet wird. Wir empfehlen Ihnen, einen CloudWatch Alarm zu erstellen, der aktiv wird, wenn der `HeapMemoryAfterGC` Anstieg über 60% liegt.

Die Maßnahmen, die Sie ergreifen können, um die Speichernutzung zu verringern, sind unterschiedlich. Sie hängen davon ab, wie Sie Apache Kafka konfigurieren. Wenn Sie beispielsweise die transaktionale Nachrichtenzustellung verwenden, können Sie den `transactional.id.expiration.ms`-Wert in Ihrer Apache-Kafka-Konfiguration von `604800000` ms auf `86400000` ms (von 7 Tagen auf 1 Tag) verringern. Dadurch wird der Speicherbedarf jeder Transaktion verringert.

Keine Nicht-MSK-Broker hinzufügen

Wenn Sie bei ZooKeeper basierten Clustern ZooKeeper Apache-Befehle zum Hinzufügen von Brokern verwenden, werden diese Broker nicht zu Ihrem MSK-Cluster hinzugefügt, und ZooKeeper Ihr Apache enthält falsche Informationen über den Cluster. Dies kann zu Datenverlust führen. Informationen zu unterstützten Clustervorgängen finden Sie unter Funktionsweise.

Aktivieren der Verschlüsselung während der Übertragung

Informationen zur Verschlüsselung während der Übertragung und zum Aktivieren dieser Verschlüsselung finden Sie unter the section called “Verschlüsselung während der Übertragung”.

Neuzuweisung von Partitionen

Um Partitionen in verschiedene Broker im selben Cluster zu verschieben, können Sie das Tool zur Neuzuweisung von Partitionen mit dem Namen `kafka-reassign-partitions.sh` verwenden. Nachdem Sie beispielsweise neue Broker hinzugefügt haben, um einen Cluster zu erweitern oder Partitionen zu verschieben, um Broker zu entfernen, können Sie diesen Cluster neu verteilen, indem Sie den neuen Brokern Partitionen neu zuweisen. Informationen zum Hinzufügen von Brokern zu einem Cluster finden Sie unter the section called “Einen Cluster erweitern”. Informationen zum Entfernen von Brokern aus einem Cluster finden Sie unter. the section called “Entfernen Sie einen Broker” Informationen zum Tool zur Neuzuweisung von Partitionen finden Sie unter Expanding your cluster in der Apache Kafka-Dokumentation.

Dokumentverlauf für das Amazon-MSK-Entwicklerhandbuch

In der folgenden Tabelle sind wichtige Änderungen am Amazon-MSK-Entwicklerhandbuch beschrieben.

Letzte Aktualisierung der Dokumentation: 25. Juni 2024

Änderung	Beschreibung	Datum
Die Graviton-Upgrade-In-Place-Funktion wurde hinzugefügt.	Sie können die Größe Ihres Cluster-Brokers von M5 oder T3 auf M7g oder von M7g auf M5 aktualisieren.	2024-6-25
3.4.0 Das Ende des Supports wurde bekannt gegeben.	Das Ende des Supports für Apache Kafka Version 3.4.0 ist der 17. Juni 2025.	24.06.2024
Funktion zum Entfernen von Brokern hinzugefügt.	Sie können die Speicher- und Rechenkapazität Ihres bereitgestellten Clusters reduzieren, indem Sie Gruppen von Brokern entfernen, ohne dass dies Auswirkungen auf die Verfügbarkeit, das Risiko der Datenbeständigkeit oder eine Unterbrechung Ihrer Datenstreaming-Anwendungen hat.	16.05.2024-
`WriteDataIdempotently` hinzugefügt zu AWSMSKReplicatorExecutionRole	WriteDataIdempotently Der AWSMSKReplicatorExecutionRole Richtlinie wurde eine Berechtigung hinzugefügt, um die Datenreplikation zwischen MSK-Clustern zu unterstützen.	2024-5-16

Änderung	Beschreibung	Datum
Graviton M7g-Broker wurden in Brasilien und Bahrain veröffentlicht.	Amazon MSK unterstützt jetzt die Verfügbarkeit von M7G-Brokern in den Regionen Südamerika (sa-east-1, São Paulo) und Naher Osten (me-south-1, Bahrain), die AWS Graviton-Prozessoren verwenden (benutzerdefiniert e ARM-basierte Prozessoren, die von Amazon Web Services entwickelt wurden).	2024-2-07
Bringen Sie Graviton M7g-Broker für die Region China auf den Markt	Amazon MSK unterstützt jetzt die Verfügbarkeit von M7G-Brokern in der Region China, die AWS Graviton-Prozessoren verwenden (kundenspezifische ARM-basierte Prozessoren, die von Amazon Web Services entwickelt wurden).	2024-01-11
Richtlinie zur Unterstützung der Amazon MSK Kafka-Ver sion	Es wurde eine Erläuterung der Support-Richtlinie für die von Amazon MSK unterstüt zte Kafka-Version hinzugefügt. Weitere Informationen finden Sie unter Apache Kafka-Ver sionen.	2023-12-08

Änderung	Beschreibung	Datum
Neue Rollenrichtlinie für die Serviceausführung zur Unterstützung von Amazon MSK Replicator.	Amazon MSK hat eine neue `AWSMSKReplicatorExecutionRole` Richtlinie zur Unterstützung von Amazon MSK Replicator hinzugefügt. Weitere Informationen finden Sie unter AWS managed policy: `AWSMSKReplicatorExecutionRole` (verwaltete Richtlinie).	2023-12-06
M7g Graviton-Unterstützung	Amazon MSK unterstützt jetzt M7G-Broker, die AWS Graviton-Prozessoren verwenden (benutzerdefinierte ARM-basierte Prozessoren, die von Amazon Web Services entwickelt wurden).	2023-11-27
Amazon MSK Replicator	Amazon MSK Replicator ist ein neues Feature, mit dem Sie Daten zwischen Amazon-MSK-Clustern replizieren können. Amazon MSK Replicator beinhaltet eine Aktualisierung der FullAccess AmazonMSK-Richtlinie. Weitere Informationen finden Sie unter AWS managed policy: `AmazonMSKFullAccess` (verwaltete Richtlinie).	2023-09-28

Änderung	Beschreibung	Datum
Für bewährte IAM-Methoden aktualisiert.	Aktualisierter Leitfaden, angepasst an die bewährten IAM-Methoden. Weitere Informationen finden Sie unter Bewährte IAM-Methoden.	2023-03-08
Updates von serviceverknüpften Rollen zur Unterstützung von privater Multi-VPC-Konnektivität	Amazon MSK umfasst jetzt AWSServiceRoleForKafka servicebezogene Rollenaktualisierungen zur Verwaltung von Netzwerkschnittstellen und VPC-Endpunkten in Ihrem Konto, sodass Cluster-Broker für Kunden in Ihrer VPC zugänglich sind. Amazon MSK verwendet Berechtigungen für `DescribeVpcEndpoints`, `ModifyVpcEndpoint` und `DeleteVpcEndpoints`. Weitere Informationen finden Sie unter Verwendung von serviceverknüpften Rollen für Amazon MSK.	2023-03-08
Unterstützung für Apache Kafka 2.7.2	Amazon MSK unterstützt jetzt Apache Kafka Version 2.7.2. Weitere Informationen finden Sie unter Unterstützte Apache Kafka-Versionen.	21.12.2021
Unterstützung für Apache Kafka 2.6.3	Amazon MSK unterstützt jetzt Apache Kafka Version 2.6.3. Weitere Informationen finden Sie unter Unterstützte Apache Kafka-Versionen.	21.12.2021

Änderung	Beschreibung	Datum
Vorabversion von MSK Serverless	MSK Serverless ist ein neues Feature, mit dem Sie Serverless-Cluster erstellen können. Weitere Informationen finden Sie unter MSK Serverless.	29.11.2021
Unterstützung für Apache Kafka 2.8.1	Amazon MSK unterstützt jetzt Apache Kafka Version 2.8.1. Weitere Informationen finden Sie unter Unterstützte Apache Kafka-Versionen.	30.09.2021
MSK Connect	MSK Connect ist ein neues Feature, mit dem Sie Apache-Kafka-Konnektoren erstellen und verwalten können. Weitere Informationen finden Sie unter MSK Connect.	16.09.2021
Unterstützung für Apache Kafka 2.7.1	Amazon MSK unterstützt jetzt Apache Kafka Version 2.7.1. Weitere Informationen finden Sie unter Unterstützte Apache Kafka-Versionen.	25.05.2021
Unterstützung für Apache Kafka 2.8.0	Amazon MSK unterstützt jetzt Apache Kafka Version 2.8.0. Weitere Informationen finden Sie unter Unterstützte Apache Kafka-Versionen.	28.04.2021

Änderung	Beschreibung	Datum
Unterstützung für Apache Kafka 2.6.2	Amazon MSK unterstützt jetzt Apache Kafka Version 2.6.2. Weitere Informationen finden Sie unter Unterstützte Apache Kafka-Versionen.	28.04.2021
Support für die Aktualisierung des Brokertyps	Sie können jetzt den Brokertyp für einen vorhandenen Cluster ändern. Weitere Informationen finden Sie unter Aktualisierung der Broker-Größe.	21. Januar 2021
Unterstützung für Apache Kafka 2.6.1	Amazon MSK unterstützt jetzt Apache Kafka Version 2.6.1. Weitere Informationen finden Sie unter Unterstützte Apache Kafka-Versionen.	19.01.2021
Unterstützung für Apache Kafka 2.7.0	Amazon MSK unterstützt jetzt Apache Kafka Version 2.7.0. Weitere Informationen finden Sie unter Unterstützte Apache Kafka-Versionen.	29. Dezember 2020

Änderung	Beschreibung	Datum
Keine neuen Cluster mit Apache Kafka Version 1.1.1	Mit Apache Kafka Version 1.1.1 können Sie keinen neuen Amazon-MSK-Cluster mehr erstellen. Wenn Sie jedoch über bestehende MSK-Cluster verfügen, auf denen Apache Kafka Version 1.1.1 ausgeführt wird, können Sie weiterhin alle derzeit unterstützten Funktionen auf diesen vorhandenen Clustern verwenden. Weitere Informationen finden Sie unter Apache-Kafka-Versionen.	24.11.2020
Metriken zur Verbraucher-Verzögerung	Amazon MSK bietet jetzt Metriken, mit denen Sie die Verzögerung von Verbrauchern überwachen können. Weitere Informationen finden Sie unter Überwachung eines Amazon-MSK-Clusters.	23.11.2020
Unterstützung für Cruise Control	Amazon MSK unterstützt LinkedIn jetzt Cruise Control. Weitere Informationen finden Sie unter Verwenden von LinkedIn's Cruise Control für Apache Kafka mit Amazon MSK.	17.11.2020

Änderung	Beschreibung	Datum
Unterstützung für Apache Kafka 2.6.0	Amazon MSK unterstützt jetzt Apache Kafka Version 2.6.0. Weitere Informationen finden Sie unter Unterstützte Apache Kafka-Versionen.	2020-10-21
Unterstützung für Apache Kafka 2.5.1	Amazon MSK unterstützt jetzt Apache Kafka Version 2.5.1. Mit Apache Kafka Version 2.5.1 unterstützt Amazon MSK die Verschlüsselung bei der Übertragung zwischen Clients und Endpunkten. ZooKeeper Weitere Informationen finden Sie unter Unterstützte Apache Kafka-Versionen.	2020-09-30
Automatische Erweiterung der Anwendung	Sie können Amazon Managed Streaming für Apache Kafka so konfigurieren, dass der Speicher Ihres Clusters bei steigender Nutzung automatis ch erweitert wird. Weitere Informationen finden Sie unter Auto Scaling.	30.09.2020
Support für Benutzername- und Passwortsicherheit	Amazon MSK unterstützt jetzt die Anmeldung bei Clustern mit einem Benutzernamen und einem Passwort. Amazon MSK speichert Anmeldein formationen in AWS Secrets Manager. Weitere Informati onen finden Sie unter SASL/ SCRAM-Authentifizierung.	2020-09-17

Änderung	Beschreibung	Datum
Unterstützung für die Aktualisierung der Apache-Kafka-Version eines Amazon-MSK-Clusters	Sie können jetzt die Apache-Kafka-Version eines vorhandenen MSK-Clusters aktualisieren.	2020-05-28
Unterstützung für Broker-Knoten vom Typ T3.small	Amazon MSK unterstützt jetzt das Erstellen von Clustern mit Brokern vom Amazon-EC2-Typ T3.small.	2020-04-08
Unterstützung von Apache Kafka 2.4.1	Amazon MSK unterstützt jetzt Apache Kafka Version 2.4.1.	02.04.2020
Unterstützung für Stream-Broker-Protokolle	Amazon MSK kann jetzt CloudWatch Broker-Protokolle an Logs, Amazon S3 und Amazon Data Firehose streamen. Firehose kann diese Protokolle wiederum an die von ihm unterstützten Ziele wie OpenSearch Service weiterleiten.	25.02.2020
Unterstützung von Apache Kafka 2.3.1	Amazon MSK unterstützt jetzt Apache Kafka Version 2.3.1.	19.12.2019
Offene Überwachung	Amazon MSK unterstützt jetzt die offene Überwachung mit Prometheus.	04.12.2019
Unterstützung von Apache Kafka 2.2.1	Amazon MSK unterstützt jetzt Apache Kafka Version 2.2.1.	31.07.2019

Änderung	Beschreibung	Datum
Allgemeine Verfügbarkeit	Zu den neuen Funktionen gehören Markierungsunterstützung, Authentifizierung, TLS-Verschlüsselung, Konfigurationen und die Möglichkeit, Broker-Speicher zu aktualisieren.	30.05.2019
Unterstützung von Apache Kafka 2.1.0	Amazon MSK unterstützt jetzt Apache Kafka Version 2.1.0.	05.02.2019

AWS Glossar

Die neueste AWS Terminologie finden Sie im AWS Glossar in der AWS-Glossar Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.