aws

Guía para desarrolladores

Transmisión gestionada de Amazon para Apache Kafka

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Transmisión gestionada de Amazon para Apache Kafka: Guía para desarrolladores

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

Bienvenido 1
¿Qué es Amazon MSK? 1
Configuración 3
Inscríbase en AWS 3
Descargar bibliotecas y herramientas 3
Aprovisionado por MSK 5
Introducción 5
Creación de un clúster 6
Creación de un rol de IAM 7
Creación de un equipo cliente 10
Crear un tema 12
Producción y consumo de datos 17
Ver las métricas de 18
Eliminación de los recursos del tutorial 19
Funcionamiento 19
Administre su clúster aprovisionado 20
Creación de un clúster 21
Enumerar clústeres 32
Conexión a un clúster de MSK 33
Obtenga los corredores de bootstrap 55
Supervisión de un clúster 56
Actualice la seguridad del clúster 103
Amplíe un clúster 106
Eliminación de un agente 109
Actualice el tamaño del agente del clúster 114
Utilice el control de crucero 117
Actualización de la configuración de un clúster 123
Reinicio de un agente para un clúster de Amazon MSK 126
Etiquetar un clúster 128
Migración a un clúster de Amazon MSK 130
Eliminar un clúster 135
Características y conceptos clave 136
Tipos de agente 137
Tamaños de corredores 140

Administrar el almacenamiento ... 142
Seguridad ... 163
Configuración de bróker ... 233
Aplicación de parches ... 308
El agente no está en línea y el cliente realiza una conmutación por error ... 310
Registro de Amazon MSK ... 313
Administración de metadatos ... 320
Recursos ... 324
Versiones de Apache Kafka ... 325
Solución de problemas del clúster de Amazon MSK ... 339
Prácticas recomendadas ... 349
Prácticas recomendadas para agentes ... 349
Mejores prácticas para los corredores de Express ... 358
Prácticas recomendadas para los clientes de Apache Kafka ... 364
MSK sin servidor ... 371
Uso de clústeres sin servidor de MSK ... 372
Creación de un clúster ... 372
Creación de un rol de IAM ... 374
Creación de un equipo cliente ... 376
Crear un tema ... 378
Producción y consumo de datos ... 379
Eliminación de recursos ... 380
Configuración ... 381
Monitorización ... 383
MSK Connect ... 386
Beneficios de Amazon MSK Connect ... 386
Introducción ... 388
Configuración de los recursos necesarios para MSK Connect ... 388
Creación de un complemento personalizado ... 393
Creación de un equipo cliente y un tema de Apache Kafka ... 394
Crea conector ... 397
Envío de datos al clúster de MSK ... 398
Descripción de los conectores ... 398
Descripción de la capacidad de los conectores ... 399
Crear un conector ... 400
Actualizar un conector ... 402

Conexión desde conectores ... 403
Creación de complementos personalizados ... 403
Descripción de los procesos de trabajo de MSK Connect ... 404
Configuración predeterminada del proceso de trabajo ... 405
Propiedades de configuración de proceso de trabajo compatibles ... 405
Creación de una configuración personalizada ... 407
Administración de los desplazamientos de los conectores ... 407
Proveedores de configuración ... 411
Consideraciones ... 412
Creación de un complemento personalizado y carga a S3 ... 412
Configuración de los parámetros y permisos para los distintos proveedores ... 414
Creación de una configuración personalizada de proceso de trabajo ... 419
Creación del conector ... 420
Roles y políticas de IAM ... 420
Descripción del rol de ejecución de servicios ... 421
Ejemplo de política de ... 424
Prevención de los problemas del suplente confuso entre servicios ... 426
AWS políticas gestionadas ... 427
Uso de roles vinculados a servicios ... 431
Habilitar el acceso a Internet ... 433
Configuración de una puerta de enlace NAT ... 433
Descripción de los nombres de host de DNS privados ... 435
Configuración de una opción de DHCP de la VPC ... 436
Configuración de atributos de DNS ... 437
Solución de errores en la creación de conectores ... 437
Seguridad ... 438
Registro ... 438
Impedir que los secretos aparezcan en los registros de los conectores ... 439
Monitorización ... 440
Ejemplos ... 443
Configuración de un conector de recepción de Amazon S3 ... 443
Configure el conector del colector EventBridge Kafka ... 445
Uso del conector de origen Debezium ... 451
Migración a Amazon MSK Connect ... 462
Descripción de los temas internos que utiliza Kafka Connect ... 463
Administración de estados ... 464

Migración de los conectores de origen ... 464
Migración de los conectores de recepción ... 465
Solución de problemas ... 467
Replicador MSK ... 468
Funcionamiento del Replicador Amazon MSK ... 469
Replicación de los datos ... 469
Replicación de metadatos ... 470
Configuración del nombre del tema ... 472
Configuración de clústeres de origen y destino ... 474
Preparación del clúster de origen de Amazon MSK ... 474
Preparación del clúster de destino de Amazon MSK ... 477
Tutorial: Creación de un Replicador Amazon MSK ... 477
Consideraciones sobre la creación de un Replicador Amazon MSK ... 478
Creación de un Replicador con la consola de AWS ... 482
Editar la configuración del Replicador MSK ... 490
Eliminar un Replicador MSK ... 491
Supervisar la replicación ... 492
Métricas del Replicador MSK ... 492
Uso de la replicación para aumentar la resiliencia ... 504
Consideraciones para crear aplicaciones Apache Kafka de varias regiones ... 504
Uso de una topología de clúster activo-activo frente a una activo-pasivo ... 505
Creación de un clúster de Kafka activo-pasivo ... 505
Conmutación por error a la región secundaria ... 506
Realización de una conmutación por error planificada ... 506
Realización de una conmutación por error no planificada ... 507
Realice una conmutación por recuperación ... 509
Creación de una configuración activa-activa ... 511
Migración de un clúster de Amazon MSK a otro ... 513
Migración de una versión autoadministrada al MirrorMaker Replicador MSK ... 513
Solución de problemas del Replicador MSK ... 514
El estado del Replicador MSK pasa de EN CREACIÓN a ERROR ... 514
El Replicador MSK aparece atascado en el estado EN CREACIÓN ... 515
El Replicador MSK no replica los datos o solo replica datos parciales ... 515
Los desplazamientos de mensajes en el clúster de destino son diferentes a las del clúster de origen ... 516

El Replicador MSK no sincroniza los desplazamientos de los grupos de consumidores o el grupo de consumidores no existe en el clúster de destino ... 516
La latencia de replicación es alta o sigue aumentando ... 517
Uso de métricas ReplicatorFailure ... 519
Prácticas recomendadas para utilizar el Replicador MSK ... 525
Administración del rendimiento del Replicador MSK mediante cuotas de Kafka ... 525
Establecimiento del periodo de retención del clúster ... 526
Integraciones de MSK ... 527
Conector de Athena para Amazon MSK ... 527
Integración de Redshift en Amazon MSK ... 527
Integración de Firehose en Amazon MSK ... 528
Tubos de acceso EventBridge ... 528
Kafka hace streaming con Express Brokers y MSK Serverless ... 530
Creación de una aplicación Kafka Streams ... 531
Planos de incrustación vectorial en tiempo real ... 534
Registro y observabilidad ... 535
Notas antes de habilitar los planos de incrustación vectorial en tiempo real ... 536
Implemente un plan de vectorización de datos en streaming ... 537
Cuota ... 541
Solicitar un aumento de cuota en Amazon MSK ... 541
Cuota de bróker estándar ... 542
Cuota de corredor exprés ... 544
Limita el rendimiento de Express Broker según el tamaño del bróker ... 547
Cuotas del Replicador MSK ... 548
Cuota para clústeres sin servidor ... 549
Cuota de MSK Connect ... 551
Historial de documentos ... 552
... dlxii

Le damos la bienvenida a la Guía para desarrolladores de Amazon MSK

Bienvenido a la guía para desarrolladores de Amazon Managed Streaming for Apache Kafka. Los temas siguientes pueden ser de ayuda para comenzar a utilizar esta guía, en función de lo que intente hacer.

- Cree un clúster aprovisionado de MSK siguiendo el tutorial. Introducción a Amazon MSK
- Profundice en las funciones de MSK Provisioned en. ¿Qué es MSK Provisioned?
- Ejecute Apache Kafka sin tener que administrar ni escalar la capacidad del clúster con MSK Serverless.
- Use MSK Connect para transmitir datos hacia y desde su clúster de Apache Kafka.
- Utilice MSK Replicator para replicar de forma fiable los datos en clústeres aprovisionados por MSK en diferentes clústeres o en el mismo lugar. Regiones de AWS

Para los aspectos destacados, los detalles del producto y los precios, consulte la página de servicio de Amazon MSK.

¿Qué es Amazon MSK?

Amazon Managed Streaming for Apache Kafka (Amazon MSK) es un servicio totalmente administrado que permite crear y ejecutar aplicaciones que utilizan Apache Kafka para procesar datos de streaming. Amazon MSK proporciona las operaciones de plano de control, como las de creación, actualización y eliminación de clústeres. Le permite utilizar operaciones de plano de datos de Apache Kafka, como producir y consumir datos. Ejecuta versiones de código abierto de Apache Kafka. Esto significa que las aplicaciones, herramientas y complementos existentes de los socios y la comunidad Apache Kafka son compatibles sin necesidad de cambios en el código de la aplicación. Puede utilizar Amazon MSK para crear clústeres que utilicen cualquiera de las versiones de Apache Kafka que figuran en la siguiente lista de the section called “Versiones compatibles de Apache Kafka”.

Estos componentes describen la arquitectura de Amazon MSK:

- Nodos de intermediario: al crear un clúster de Amazon MSK, debe especificar cuántos nodos de intermediario desea que Amazon MSK cree en cada zona de disponibilidad. El mínimo es un

agente por zona de disponibilidad. Cada zona de disponibilidad tiene su propia subred de nube virtual privada (VPC).

Amazon MSK Provisioned ofrece dos tipos de agentes: Agentes estándar de Amazon MSK y. Agentes de Amazon MSK Express En MSK Serverless, MSK administra los nodos intermediarios que se utilizan para gestionar el tráfico y usted solo aprovisiona los recursos del servidor Kafka a nivel de clúster.

- ZooKeeper nodos: Amazon MSK también crea los ZooKeeper nodos de Apache por usted. Apache ZooKeeper es un servidor de código abierto que permite una coordinación distribuida de gran fiabilidad.
- KRaft controladores: la comunidad de Apache Kafka se desarrolló KRaft para reemplazar a Apache ZooKeeper para la administración de metadatos en los clústeres de Apache Kafka. En el KRaft modo, los metadatos del clúster se propagan dentro de un grupo de controladores de Kafka, que forman parte del clúster de Kafka, en lugar de propagarse entre nodos. ZooKeeper KRaftLos controladores se incluyen sin coste adicional para usted y no requieren ninguna configuración o administración adicionales por su parte.
- Productores, consumidores y creadores de temas: Amazon MSK le permite utilizar operaciones de plano de datos de Apache Kafka para crear temas y para producir y consumir datos.
- Operaciones de clúster Puede usar el AWS Management Console, el AWS Command Line Interface (AWS CLI) o el del APIs SDK para realizar operaciones del plano de control. Por ejemplo, puede crear o eliminar un clúster de Amazon MSK, mostrar todos los clústeres de una cuenta, ver las propiedades de un clúster y actualizar el número y el tipo de agentes de un clúster.

Amazon MSK detecta los escenarios de error más comunes para clústeres, y se recupera de dichos escenarios, de modo que las aplicaciones de productoras y consumidoras puedan continuar sus operaciones de escritura y lectura con un impacto mínimo. Cuando Amazon MSK detecta un error de agente, mitiga el error o reemplaza al agente inaccesible o incorrecto por uno nuevo. Además, cuando es posible, reutiliza el almacenamiento del agente más antiguo para reducir los datos que Apache Kafka necesita replicar. El impacto a la disponibilidad se limita al tiempo necesario para que Amazon MSK complete la detección y recuperación. Después de una recuperación, las aplicaciones de productor y consumidor pueden seguir comunicándose con las mismas direcciones IP del agente que usaban antes del error.

Configuración de Amazon MSK

Antes de usar Amazon MSK por primera vez, finalice las siguientes tareas.

Tareas

- Inscríbase en AWS
- Descargar bibliotecas y herramientas

Inscríbase en AWS

Cuando te registras AWS, tu cuenta de Amazon Web Services se registra automáticamente en todos los servicios de Amazon AWS, incluido Amazon MSK. Solo se le cobrará por los servicios que utilice.

Si ya tienes una AWS cuenta, pasa a la siguiente tarea. Si no dispone de una cuenta de AWS , utilice el siguiente procedimiento para crear una.

Creación de una cuenta de Amazon Web Services

1. Abre el https://portal.aws.amazon.com/billing/registro.
2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro consiste en recibir una llamada telefónica o mensaje de texto e indicar un código de verificación en el teclado del teléfono.

   Cuando te registras en un Cuenta de AWS, Usuario raíz de la cuenta de AWSse crea un. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar tareas que requieren acceso de usuario raíz.

Descargar bibliotecas y herramientas

Las siguientes bibliotecas y herramientas pueden ayudarlo a trabajar con Amazon MSK:

- La AWS Command Line Interface (AWS CLI) es compatible con Amazon MSK. AWS CLI Le permite controlar varios Amazon Web Services desde la línea de comandos y automatizarlos mediante scripts. Actualice su versión AWS CLI a la última para asegurarse de que es compatible con las funciones de Amazon MSK que se documentan en esta guía del usuario.

Para instrucciones detalladas sobre cómo actualizar la AWS CLI, consulte Instalación de la AWS Command Line Interface. Después de instalar el AWS CLI, debe configurarlo. Para obtener información sobre cómo configurar el AWS CLI, consulte aws configure.

- La referencia de la API de Amazon Managed Streaming para Kafka documenta las operaciones de la API compatibles con Amazon MSK.
- Amazon Web Services SDKs for Go, Java, .NET JavaScript, Node.js, PHP, Python y Ruby incluyen soporte y ejemplos de Amazon MSK.

¿Qué es MSK Provisioned?

Los clústeres aprovisionados de Amazon MSK ofrecen una amplia gama de características y capacidades para ayudarlo a optimizar el rendimiento del clúster y satisfacer sus necesidades de streaming. En los temas siguientes se describe la funcionalidad en detalle.

MSK Provisioned es una opción de despliegue de clústeres de MSK que permite configurar y escalar manualmente los clústeres de Apache Kafka. Esto le proporciona distintos niveles de control sobre la infraestructura que alimenta su entorno de Apache Kafka. Con MSK Provisioned, puede elegir los tipos de instancias, los volúmenes de almacenamiento (agentes estándar) y el número de nodos de agente que componen sus clústeres de Kafka. También puede escalar su clúster añadiendo o eliminando agentes a medida que evolucionen sus necesidades de procesamiento de datos. Esta flexibilidad le permite optimizar los clústeres para sus requisitos de carga de trabajo específicos, ya sea para maximizar el rendimiento, la capacidad de retención u otras características de rendimiento. Además de las opciones de configuración de la infraestructura, MSK Provisioned ofrece beneficios operativos, de supervisión y de seguridad de nivel empresarial. Esto incluye funciones como las actualizaciones de las versiones de Apache Kafka, la seguridad integrada mediante el cifrado y el control de acceso, y la integración con otras, Servicios de AWS como Amazon, CloudWatch para la supervisión. MSK Provisioned ofrece dos tipos principales de corredores: Standard y Express.

Para obtener información sobre la API aprovisionada de MSK, consulte la referencia de la API de Amazon MSK.

Introducción a Amazon MSK

En este tutorial, se muestra un ejemplo de cómo crear un clúster de MSK, cómo producir y consumir datos y cómo supervisar el estado del clúster mediante métricas. Este ejemplo no representa todas las opciones que puede elegir al crear un clúster de MSK. En diferentes partes de este tutorial, elegimos opciones predeterminadas por motivos de simplicidad. Esto no significa que sean las únicas opciones que funcionan para configurar un clúster de MSK o instancias de cliente.

Temas

- Paso 1: Crear un clúster aprovisionado de MSK
- Paso 2: creación de un rol de IAM que otorgue acceso a la creación de temas en el clúster de Amazon MSK
- Paso 3: creación de un equipo cliente

- Paso 4: creación de un tema en el clúster de Amazon MSK
- Paso 5: producción y consumo de datos
- Paso 6: Usa Amazon CloudWatch para ver las métricas de Amazon MSK
- Paso 7: Elimine los AWS recursos creados para este tutorial

Paso 1: Crear un clúster aprovisionado de MSK

En este paso de Introducción a Amazon MSK, debe crear un clúster aprovisionado de Amazon MSK. Utilice la opción de creación rápida de la AWS Management Console para crear este clúster.

Para crear un clúster de Amazon MSK mediante AWS Management Console

1. ¿Iniciar sesión en la AWS Management Console consola Amazon MSK y abrirla desde https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. Elija Create cluster.
3. En Método de creación, deje seleccionada la opción Creación rápida. La opción Creación rápida le permite crear un clúster con la configuración predeterminada.
4. En Nombre del clúster, ingrese un nombre para el clúster. Por ejemplo, **`MSKTutorialCluster`**.
5. Para las propiedades generales del clúster, haga lo siguiente:
   a. En Tipo de clúster, seleccione Aprovisionado.
   b. Elija una versión de Apache Kafka para ejecutarla en los corredores. Seleccione Ver compatibilidad de versiones para ver una tabla comparativa.
   c. En el tipo de corredor, elija corredores estándar o exprés.
   d. Elija un tamaño de bróker.
6. En la tabla que aparece debajo de Todas las configuración del clúster, copie los valores de las siguientes configuraciones y guárdalos, ya que los necesitará más adelante en este tutorial:
   - VPC
   - Subredes
   - Grupos de seguridad asociados con la VPC
7. Elija Create cluster.
8. Compruebe el estado del clúster en Estado, en la página Resumen del clúster. El estado cambia de En creación a Activo a medida que Amazon MSK aprovisiona el clúster. Cuando el estado

sea Activo, puede conectarse al clúster. Para obtener más información acerca del estado de un clúster, consulte Conozca los estados de los clústeres aprovisionados por MSK.

Paso siguiente

Paso 2: creación de un rol de IAM que otorgue acceso a la creación de temas en el clúster de Amazon MSK

Paso 2: creación de un rol de IAM que otorgue acceso a la creación de temas en el clúster de Amazon MSK

En este paso, se realizan dos tareas. La primera tarea consiste en crear una política de IAM que conceda acceso para crear temas en el clúster y enviarles datos. La segunda tarea consiste en crear un rol de IAM y asociarle esta política. En un paso posterior, se crea un equipo cliente que asume este rol y lo utiliza para crear un tema en el clúster y enviar datos a ese tema.

Creación de una política de IAM que permita crear temas y escribir en ellos

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, seleccione Políticas.
3. Elija Crear política.
4. En el editor de políticas, selecciona JSON y, a continuación, sustituye el JSON de la ventana del editor por el siguiente JSON.

   En el ejemplo siguiente, sustituya lo siguiente:

   - *region*con el código del Región de AWS lugar donde creó el clúster.
   - *Account-ID*con tu Cuenta de AWS ID.
   - *MSKTutorialCluster*y*MSKTutorialCluster*/*7d7131e1-25c5-4e9a-9ac5-ea85bee4da11-14*, con el nombre del clúster y su ID.

```
{
    "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/7d7131e1-25c5-4e9a-9ac5-ea85bee4da11-14"
            ]
        },
        {
            "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/*"
            ]
        }
    ]
}
```

Para obtener instrucciones sobre cómo escribir políticas de seguridad, consultethe section called "Control de acceso de IAM".

5. Elija Siguiente.
6. En la página Revisar y crear, haga lo siguiente:

   a. En el nombre de la política, introduzca un nombre descriptivo, como**msk-tutorial-policy**.

   b. En Permisos definidos en esta política, revise y and/or edite los permisos definidos en su política.

c. (Opcional) Para ayudar a identificar, organizar o buscar la política, elija Agregar nueva etiqueta para agregar etiquetas como pares clave-valor. Por ejemplo, agrega una etiqueta a tu política con el par clave-valor de y. **`Environment Test`**

   Para obtener más información sobre el uso de etiquetas, consulte Etiquetas para AWS Identity and Access Management recursos en la Guía del usuario de IAM.

7. Elija Crear política.

Creación de un rol de IAM y asociarle la política

1. En el panel de navegación, elija Funciones y, a continuación, elija Crear función.
2. En la página Seleccionar entidad de confianza, haga lo siguiente:

   a. En Tipo de entidad de confianza, elija Servicio de AWS.

   b. En Servicio o caso de uso, elija EC2.

   c. En Use case (Caso de uso), elija EC2.

3. Elija Siguiente.
4. En la página Agregar permisos, haga lo siguiente:

   a. En el cuadro de búsqueda de Políticas de permisos, introduce el nombre de la política que creaste anteriormente para este tutorial. A continuación, seleccione la casilla situada a la izquierda del nombre de la política.

   b. (Opcional) Configure un límite de permisos. Se trata de una característica avanzada que está disponible para los roles de servicio, pero no para los roles vinculados a servicios. Para obtener información sobre cómo establecer un límite de permisos, consulte Crear funciones y adjuntar políticas (consola) en la Guía del usuario de IAM.

5. Elija Siguiente.
6. En la página Nombrar, revisar y crear, haga lo siguiente:

   a. En el nombre del rol, introduzca un nombre descriptivo, como. **`msk-tutorial-role`**

   > **Important**
   >
   > Cuando asigne un nombre a un rol, tenga en cuenta lo siguiente:
   >
   > - Los nombres de los roles deben ser únicos dentro de su Cuenta de AWS perfil y no pueden ser únicos por mayúsculas y minúsculas.

Por ejemplo, no puedes crear roles denominados tanto **`PRODROLE`** como **`prodrole`**. Cuando se utiliza un nombre de rol en una política o como parte de un ARN, el nombre de rol distingue entre mayúsculas y minúsculas, sin embargo, cuando un nombre de rol les aparece a los clientes en la consola, como por ejemplo durante el proceso de inicio de sesión, el nombre de rol no distingue entre mayúsculas y minúsculas.

- Dado que otras entidades podrían hacer referencia al rol, no es posible editar el nombre del rol una vez creado.

b. (Opcional) En Descripción, ingrese una descripción para el rol.

c. (Opcional) Para editar los casos de uso y los permisos del rol, en el paso 1: seleccionar entidades de confianza o en el paso 2: agregar secciones de permisos, seleccione Editar.

d. (Opcional) Para ayudar a identificar, organizar o buscar el rol, elija Agregar nueva etiqueta para agregar etiquetas como pares de valores clave. Por ejemplo, agrega una etiqueta a tu rol con el par clave-valor de y. **`ProductManager John`**

Para obtener más información sobre el uso de etiquetas, consulte Etiquetas para AWS Identity and Access Management recursos en la Guía del usuario de IAM.

7. Revise el rol y, a continuación, elija Crear rol.

Paso siguiente

Paso 3: creación de un equipo cliente

Paso 3: creación de un equipo cliente

En este paso de Introducción a Amazon MSK, creará un equipo cliente. Utilice este equipo cliente para crear un tema que produzca y consuma datos. Para simplificar, creará este equipo cliente en la VPC asociada al clúster de MSK para que el cliente pueda conectarse fácilmente al clúster.

Creación de un equipo cliente

1. Abre la EC2 consola de Amazon en https://console.aws.amazon.com/ec2/.
2. En el panel de control de la EC2 consola de Amazon, selecciona Launch instance.
3. En Nombre y etiquetas, en Nombre, introduce un nombre descriptivo para el equipo cliente de forma que puedas rastrearlo fácilmente. Por ejemplo, **`MSKTutorialClient`**.

4. En Imágenes de aplicaciones y sistema operativo (Amazon Machine Image), para Amazon Machine Image (AMI), elija Amazon Linux 2 AMI (HVM): Kernel 5.10, tipo de volumen SSD.
5. En el tipo de instancia, mantenga la selección predeterminada de t2.micro.
6. En Par de claves (inicio de sesión), elija un par de claves existente o cree uno nuevo. Si no necesita un par de claves para conectarse a la instancia, puede elegir Continuar sin un par de claves (no se recomienda).

   Para crear un nuevo par de claves, haga lo siguiente:

   a. Selecciona Crear nuevo key pair.
   b. En Key pair name (Nombre del par de claves), ingrese **`MSKKeyPair`**.
   c. Para el tipo de par de claves y el formato de archivo de clave privada, mantenga las selecciones predeterminadas.
   d. Elija Crear par de claves.

   También puede utilizar un par de claves existente.
7. Desplácese hacia abajo en la página y expanda la sección de detalles avanzados y, a continuación, haga lo siguiente:

   - Para el perfil de instancia de IAM, elija una función de IAM que desee que asuma la máquina cliente.

     Si no tiene una función de IAM, haga lo siguiente:

     i. Selecciona Crear un nuevo perfil de IAM.
     ii. Realice los pasos mencionados en el paso 2: Cree un rol de IAM.
8. Seleccione Iniciar instancia.
9. Elija View Instances (Ver instancias). A continuación, en la columna Grupos de seguridad, elija el grupo de seguridad asociado a la nueva instancia. Copie el ID del grupo de seguridad y guárdelo para más adelante.
10. Abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc/.
11. En el panel de navegación, elija Grupos de seguridad. Busque el grupo de seguridad cuyo ID guardó en the section called “Creación de un clúster”.
12. En la pestaña Reglas de entrada, elija Editar reglas de entrada.
13. Seleccione Agregar regla.

14. En la nueva regla, elija All traffic (Todo el tráfico) en la columna Type (Tipo). En el segundo campo de la columna Origen escriba el ID del grupo de seguridad del equipo cliente. Este es el grupo cuyo nombre guardó después de lanzar la instancia de la máquina cliente.
15. Seleccione Guardar reglas. Ahora, el grupo de seguridad del clúster puede aceptar el tráfico que proviene del grupo de seguridad de la máquina cliente.

Paso siguiente

Paso 4: creación de un tema en el clúster de Amazon MSK

Paso 4: creación de un tema en el clúster de Amazon MSK

En este paso de Introducción a Amazon MSK, debe instalar las bibliotecas y herramientas del cliente de Apache Kafka en el equipo cliente y, a continuación, creer un tema.

> **Warning**
>
> Los números de versión de Apache Kafka utilizados en este tutorial son solo ejemplos. Se recomienda utilizar la misma versión del cliente que la versión de clúster de MSK. Es posible que a una versión de cliente anterior le falten ciertas funciones y correcciones de errores críticos.

Temas

- Determinar la versión del clúster de MSK
- Crear un tema en la máquina cliente

Determinar la versión del clúster de MSK

1. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/.
2. En la barra de navegación, elige la región en la que creaste el clúster de MSK.
3. Elija el clúster de MSK.
4. Anote la versión de Apache Kafka utilizada en el clúster.
5. Sustituya las instancias de los números de versión de Amazon MSK de este tutorial por la versión obtenida en el paso 3.

Crear un tema en la máquina cliente

1. Conéctese a su máquina cliente.

   a. Abre la EC2 consola de Amazon en https://console.aws.amazon.com/ec2/.

   b. En el panel de navegación, seleccione Instances (Instancias). A continuación, seleccione la casilla de verificación situada junto al nombre de la máquina cliente en la que creóPaso 3: creación de un equipo cliente.

   c. Elija Actions (Acciones) y, a continuación, elija Connect (Conectar). Siga las instrucciones de la consola para conectarse al equipo cliente.

2. Instale Java y configure la variable de entorno de la versión de Kafka.

   a. Instale Java en la máquina cliente ejecutando el siguiente comando.

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

   b. Guarde la versión de Kafka del clúster de MSK en la variable de entornoKAFKA_VERSION, como se muestra en el siguiente comando. Necesitará esta información durante toda la configuración.

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

   Por ejemplo, si utiliza la versión 3.6.0, utilice el siguiente comando.

   ```
   export KAFKA_VERSION=3.6.0
   ```

3. Descargue y extraiga Apache Kafka.

   a. Ejecute el siguiente comando para descargar Apache Kafka.

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

   Por ejemplo, si su clúster de MSK usa la versión 3.6.0 de Apache Kafka, ejecute el siguiente comando.

   ```
   wget https://archive.apache.org/dist/kafka/3.6.0/kafka_2.13-3.6.0.tgz
   ```

> **Note**
>
> La siguiente lista presenta información alternativa sobre la descarga de Kafka que puede utilizar si tiene algún problema.
>
> - Si tiene problemas de conectividad o desea utilizar un sitio réplica, pruebe a utilizar el selector de réplicas de Apache, tal y como se muestra en el siguiente comando.
>
>   ```
>   wget https://www.apache.org/dyn/closer.cgi?path=/kafka/$KAFKA_VERSION/kafka_2.13-$KAFKA_VERSION.tgz
>   ```
>
> - Descargue la versión adecuada directamente desde el sitio web de Apache Kafka.

b. Ejecute el siguiente comando en el directorio donde descargó el archivo TAR del paso anterior.

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

c. Guarde la ruta completa al directorio recién creado dentro de la variable de KAFKA_ROOT entorno.

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

4. Configure la autenticación para su clúster de MSK.

a. Descargue la versión más reciente del archivo JAR de IAM de Amazon MSK que se encuentra en el `$KAFKA_ROOT/libs` directorio. Use el siguiente comando para descargar el archivo y sustitúyalo por el *`{LATEST VERSION}`* número de versión real.

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

El archivo JAR de IAM de Amazon MSK permite que su máquina cliente acceda al clúster de MSK mediante la autenticación de IAM.

> **Note**
>
> Antes de ejecutar cualquier comando de Kafka que interactúe con el clúster de MSK, puede que tenga que añadir el archivo JAR de IAM de Amazon MSK a la ruta de clases de Java. Establezca la variable de `CLASSPATH` entorno, como se muestra en el siguiente ejemplo.
>
> ```
> export CLASSPATH=$KAFKA_ROOT/libs/aws-msk-iam-auth-{LATEST VERSION}-
> all.jar
> ```
>
> Esto establece el valor `CLASSPATH` para toda la sesión, haciendo que el JAR esté disponible para todos los comandos de Kafka posteriores.

b. Vaya al `$KAFKA_ROOT/config` directorio para crear el archivo de configuración del cliente.

```
cd $KAFKA_ROOT/config
```

c. Copie las siguientes configuraciones de propiedades y péguelas en un archivo nuevo. Guarde el archivo como **`client.properties`**.

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

5. (Opcional) Si tiene algún problema relacionado con la memoria o está trabajando con un gran número de temas o particiones, puede ajustar el tamaño del montón de Java para las herramientas de Kafka. Para ello, defina la variable de `KAFKA_HEAP_OPTS` entorno antes de ejecutar los comandos de Kafka.

   El siguiente ejemplo establece el tamaño máximo e inicial del montón en 512 megabytes. Ajuste estos valores en función de sus requisitos específicos y de los recursos del sistema disponibles.

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

6. Obtenga la información de conexión del clúster.

   a. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/.

b. Espere a que el estado del clúster pase a ser Activo. Esto podría tardar varios minutos. Cuando el estado pase a ser Activo, elija el nombre del clúster. Se le redirigirá a una página que contiene el resumen del clúster.

c. Seleccione Ver información del cliente.

d. Copie la cadena de conexión del punto de conexión privado.

   Obtendrá tres puntos de conexión para cada uno de los corredores. Guarde una de estas cadenas de conexión en la variable de entornoBOOTSTRAP_SERVER, como se muestra en el siguiente comando. *<bootstrap-server-string>*Sustitúyala por el valor real de la cadena de conexión.

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

7. Ejecute el siguiente comando para crear el tema.

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

Si obtiene una `NoSuchFileException` para el `client.properties` archivo, asegúrese de que este archivo existe en el directorio de trabajo actual dentro del directorio bin de Kafka.

> **Note**
>
> Si prefiere no establecer la variable de `CLASSPATH` entorno para toda la sesión, también puede anteponer la variable a cada comando de Kafka. `CLASSPATH` Este enfoque aplica la ruta de clases solo a ese comando específico.
>
> ```
> CLASSPATH=$KAFKA_ROOT/libs/aws-msk-iam-auth-{LATEST VERSION}-all.jar \
> $KAFKA_ROOT/bin/kafka-topics.sh --create \
> --bootstrap-server $BOOTSTRAP_SERVER \
> --command-config $KAFKA_ROOT/config/client.properties \
> --replication-factor 3 \
> --partitions 1 \
> --topic MSKTutorialTopic
> ```

8. (Opcional) Compruebe que el tema se haya creado correctamente.

a. Si el comando se ejecuta correctamente, debería aparecer el siguiente mensaje: `Created topic MSKTutorialTopic.`

b. Enumere todos los temas para confirmar que su tema existe.

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

Si el comando no funciona o se produce un error, consulte Solución de problemas del clúster de Amazon MSK para obtener información sobre la solución de problemas.

9. (Opcional) Si desea conservar las variables de entorno para los siguientes pasos de este tutorial, omita este paso. De lo contrario, puede desconfigurar las variables, como se muestra en el siguiente ejemplo.

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

Paso siguiente

Paso 5: producción y consumo de datos

Paso 5: producción y consumo de datos

En este paso de Introducción a Amazon MSK, producirá y consumirá datos.

Producción y consumo de mensajes

1. Ejecute el siguiente comando para iniciar un productor de la consola.

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

2. Escriba el mensaje que desee y pulse Enter (Entrar). Repita este paso dos o tres veces. Cada vez que introduzca una línea y pulse Enter (Entrar), dicha línea se envía al clúster de Apache Kafka como un mensaje separado.
3. Mantenga abierta la conexión al equipo cliente y, a continuación, abra una segunda conexión independiente a dicho equipo en una nueva ventana. Como se trata de una sesión nueva, vuelva a configurar las variables KAFKA_ROOT y de BOOTSTRAP_SERVER entorno. Para obtener

información sobre cómo configurar estas variables de entorno, consulteCrear un tema en la máquina cliente.

4. Ejecute el siguiente comando con la segunda cadena de conexión a la máquina cliente para crear un consumidor de consola.

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

   Debería empezar a ver los mensajes que introdujo anteriormente cuando utilizó el comando console producer.

5. Escriba más mensajes en la ventana del productor y observe cómo aparecen en la ventana del consumidor.

Paso siguiente

Paso 6: Usa Amazon CloudWatch para ver las métricas de Amazon MSK

Paso 6: Usa Amazon CloudWatch para ver las métricas de Amazon MSK

En este paso de Introducción a Amazon MSK, analizará las métricas de Amazon MSK en Amazon. CloudWatch

Para ver las métricas de Amazon MSK en CloudWatch

1. Abre la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas.
3. Seleccione la pestaña All metrics (Todas las métricas) y, a continuación, seleccione AWS/Kafka.
4. Para ver métricas de nivel de agente, elija Broker ID, Cluster Name (ID de agente, Nombre del clúster). En las métricas de nivel de clúster, elija Cluster Name (Nombre del clúster).
5. (Opcional) En el panel de gráficos, seleccione una estadística y un período de tiempo y, a continuación, cree una CloudWatch alarma con estos ajustes.

Paso siguiente

Paso 7: Elimine los AWS recursos creados para este tutorial

Paso 7: Elimine los AWS recursos creados para este tutorial

En el último paso de Introducción a Amazon MSK, debe eliminar el clúster de MSK y el equipo cliente que creó para este tutorial.

Para eliminar los recursos mediante el AWS Management Console

1. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/.
2. Elija el nombre del clúster. Por ejemplo, MSKTutorialClúster.
3. Seleccione Actions (Acciones) y, a continuación, seleccione Delete (Eliminar).
4. Abre la EC2 consola de Amazon en https://console.aws.amazon.com/ec2/.
5. Elija la instancia que creó para el equipo cliente, por ejemplo, **`MSKTutorialClient`**.
6. Elija Estado de la instancia y, luego, Terminar instancia.

Eliminación de la política y el rol de IAM

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, seleccione Roles.
3. En el cuadro de búsqueda, escriba el nombre del rol de IAM que creó para este tutorial.
4. Elija el rol. A continuación, elija Eliminar rol para confirmar la eliminación.
5. En el panel de navegación, seleccione Políticas.
6. En el cuadro de búsqueda, escriba el nombre de la política que creó para este tutorial.
7. Elija la política para abrir su página de resumen. En la página Resumen de la política, elija Eliminar política.
8. Elija Eliminar.

Funcionamiento de Amazon MSK

Amazon MSK es un servicio de Apache Kafka totalmente gestionado que facilita la creación y ejecución de aplicaciones que utilizan Apache Kafka para procesar datos de streaming. Esta guía proporciona información para ayudar a los desarrolladores a entender cómo funciona Amazon MSK y cómo utilizarlo eficazmente en sus aplicaciones.

En un nivel alto, Amazon MSK proporciona un clúster de Apache Kafka totalmente gestionado que es aprovisionado y operado por. AWS Esto significa que no tiene que preocuparse por el

aprovisionamiento de EC2 instancias, la configuración de la red, la administración de los agentes de Kafka o la realización de tareas de mantenimiento continuas. En su lugar, puede centrarse en crear su aplicación y dejar que Amazon MSK se encargue de la infraestructura. Amazon MSK aprovisiona automáticamente los recursos informáticos, de almacenamiento y de red necesarios, y proporciona funciones como el escalado automático, la alta disponibilidad y la conmutación por error para garantizar que su clúster de Kafka sea fiable y esté altamente disponible. Esta guía describe los componentes clave de Amazon MSK y cómo puede utilizarlos para crear aplicaciones de transmisión de datos.

Administre su clúster aprovisionado

Un clúster de Amazon MSK es el recurso principal de Amazon MSK que puede crear en su cuenta. En los temas de esta sección, se describe cómo realizar las operaciones comunes de Amazon MSK. Para obtener una lista de todas las operaciones que puede realizar en un clúster de MSK, consulte lo siguiente:

- la AWS Management Console,
- La referencia de la API de Amazon MSK
- La referencia de los comandos de la CLI de Amazon MSK

Temas

- Cree un clúster aprovisionado de MSK
- Enumeración de clústeres de Amazon MSK
- Conexión a un clúster de Amazon MSK
- Obtención de agentes de arranque para un clúster de Amazon MSK
- Supervise un clúster aprovisionado de Amazon MSK
- Actualización de la configuración de seguridad de un clúster de Amazon MSK
- Expansión de la cantidad de agentes en un clúster de Amazon MSK
- Eliminación de un agente de un clúster de Amazon MSK
- Aprovisione el rendimiento del almacenamiento para los corredores estándar en un clúster de Amazon MSK
- Actualización del tamaño del agente del clúster de Amazon MSK
- Utilice LinkedIn el control de crucero para Apache Kafka con Amazon MSK
- Actualización de la configuración de un clúster de Amazon MSK

- Reinicio de un agente para un clúster de Amazon MSK
- Etiquetado de un clúster de Amazon MSK
- Migración a un clúster de Amazon MSK
- Eliminar un clúster aprovisionado de Amazon MSK

Cree un clúster aprovisionado de MSK

> **Important**
>
> No puede cambiar la VPC de un clúster aprovisionado de MSK después de crear el clúster.

Antes de poder crear un clúster aprovisionado de MSK, debe tener una (Amazon Virtual Private Cloud VPC) y configurar subredes dentro de esa VPC.

Para los corredores estándar de la región EE.UU. Oeste (Norte de California), se necesitan dos subredes en dos zonas de disponibilidad diferentes. En el resto de las regiones donde esté disponible Amazon MSK, puede especificar dos o tres subredes. Todas las subredes deben estar en diferentes zonas de disponibilidad. Para los corredores de Express, necesita tres subredes en tres zonas de disponibilidad diferentes. Al crear un clúster aprovisionado de MSK, Amazon MSK distribuye los nodos del broker de manera uniforme en las subredes que especifique.

Temas

- Cree un clúster aprovisionado de MSK mediante el AWS Management Console
- Cree un clúster de Amazon MSK aprovisionado mediante AWS CLI
- Cree un clúster aprovisionado de MSK con una configuración de Amazon MSK personalizada mediante el AWS CLI
- Cree un clúster aprovisionado de MSK mediante la API de Amazon MSK

Cree un clúster aprovisionado de MSK mediante el AWS Management Console

Los procedimientos de este tema describen la tarea común de crear un clúster aprovisionado de MSK mediante la opción de creación personalizada de. AWS Management Console Con otras opciones disponibles en el AWS Management Console, también puede crear lo siguiente:

- Un clúster sin servidor

- Un clúster aprovisionado por MSK que utiliza la opción de creación rápida

Procedimientos de este tema

- Paso 1: Instalación y configuración iniciales del clúster
- Paso 2: Configure los ajustes de almacenamiento y clúster
- Paso 3: Configure los ajustes de red
- Paso 4: Configure los ajustes de seguridad
- Paso 5: Configurar las opciones de supervisión
- Paso 6: Revise la configuración del clúster

Paso 1: Instalación y configuración iniciales del clúster

1. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/.
2. Elija Create cluster.
3. Para el Método de creación de clústeres, elija Creación personalizada.
4. Para el nombre del clúster, especifique un nombre que sea único y que no contenga más de 64 caracteres.
5. En Tipo de clúster, seleccione Aprovisionado.
6. Para la versión de Apache Kafka, elija una versión para ejecutarla en los corredores. Para ver una comparación de las funciones de Amazon MSK compatibles con cada versión de Apache Kafka, seleccione Ver compatibilidad de versiones.
7. En la sección Agentes, haga lo siguiente:

    a. Para el tipo de corredor, elija una de las siguientes opciones:

    - Express Brokers: corredores escalables y de alto rendimiento con almacenamiento virtual totalmente gestionado. Elija este tipo de intermediario para aplicaciones exigentes y de alto rendimiento.
    - Brokers estándar: Broker tradicional de Kafka con control total de la configuración. Elija este tipo de agente para cargas de trabajo de uso general con requisitos de rendimiento moderados.

Para obtener más información sobre estos tipos de corredores, consulte. Tipos de brókeres de Amazon MSK

b. Para Broker Size, elija el tamaño que desee utilizar para el clúster en función de las necesidades de procesamiento, memoria y almacenamiento del clúster.

c. En Número de zonas, elija el número de zonas Zonas de disponibilidad de AWSen las que se distribuyen los corredores.

   Los corredores exprés requieren tres zonas de disponibilidad para una mayor disponibilidad.

d. En el caso de los corredores por zona, especifique el número de agentes que desea que Amazon MSK cree en cada zona de disponibilidad. El mínimo es un agente por zona de disponibilidad y el máximo es de 30 agentes por clúster para los clústeres ZooKeeper basados y 60 agentes por clúster para los clústeres KRaft basados.

Paso 2: Configure los ajustes de almacenamiento y clúster

Este procedimiento describe cómo puede configurar sus necesidades de almacenamiento de datos en todos los corredores y especificar el modo de almacenamiento. Esto le ayuda a definir sus requisitos de almacenamiento de datos en función de sus necesidades de carga de trabajo. Además, este procedimiento describe los ajustes de configuración del clúster que controlan el funcionamiento de sus corredores. Estos ajustes incluyen las configuraciones de los corredores, la configuración de los temas predeterminados y la política de almacenamiento por niveles.

1. Si seleccionó el tipo de agente como Estándar, haga lo siguiente en la sección Almacenamiento:

   a. En Almacenamiento, elija la cantidad inicial de almacenamiento que desea que tenga su clúster. Una vez creado el clúster, no se puede reducir la capacidad de almacenamiento.

   b. (Opcional) Según el tamaño del agente (tamaño de la instancia) que haya seleccionado, también puede especificar el rendimiento del almacenamiento aprovisionado por agente. Esta opción le permite asignar un rendimiento de entrada y salida (E/S) dedicado a los volúmenes de Amazon EBS de cada agente.

      Para activar esta opción, elija el tamaño del agente (tamaño de instancia) kafka.m5.4xlarge o superior para las instancias x86 y kafka.m7g.2xlarge o superior para las instancias basadas en Graviton. A continuación, seleccione la casilla Habilitar el rendimiento del almacenamiento aprovisionado. Al seleccionar esta casilla de verificación, puede establecer manualmente un rendimiento mínimo de 250 MiB por segundo. Esto resulta útil para

las cargas de trabajo con un uso intensivo de E/S o las aplicaciones que requieren un rendimiento de almacenamiento predecible y de alta velocidad. Para obtener más información, consulte ???.

c. En el modo de almacenamiento en clúster, especifique cómo se almacenan y administran los datos dentro del clúster. Esta opción determina el tipo y la configuración del almacenamiento que utilizan sus corredores. Seleccione una de las siguientes opciones:

- Solo almacenamiento en EBS: almacena todos los datos de los temas de forma local en los volúmenes de Amazon Elastic Block Store (Amazon EBS) adjuntos a cada agente. Elija este modo para satisfacer necesidades de rendimiento uniformes y acceder rápidamente a los mensajes recientes.
- Almacenamiento por niveles y almacenamiento de EBS: combina los datos locales de Amazon EBS con un almacenamiento remoto y rentable para conjuntos de datos de gran tamaño en Amazon S3. Este modo reduce los costos de almacenamiento de Amazon EBS, permite una retención de datos más prolongada y escala el almacenamiento automáticamente sin intervención manual. Elija este modo cuando desee conservar los datos durante períodos más prolongados a un costo menor o espere que sus necesidades de almacenamiento aumenten considerablemente.

> **Note**
>
> No necesita administrar el almacenamiento para Express Brokers.

2. Para la configuración del clúster, especifique una de las siguientes opciones para definir el comportamiento del clúster:

- Configuración predeterminada de Amazon MSK: contiene un conjunto predefinido de configuraciones optimizadas para casos de uso de uso general. Elija esta opción para una configuración e implementación rápidas del clúster. Para obtener información acerca de las configuraciones de Amazon MSK, consulte Configuración aprovisionada de Amazon MSK.
- Configuración personalizada: le permite especificar su propia configuración de corredor y tema. Puede elegir una configuración personalizada existente de la lista o crear una nueva configuración personalizada. Elija esta opción para obtener un control más preciso para sus corredores, como ajustes específicos del rendimiento, ajustes de seguridad y mucho más.

3. Elija Paso siguiente para continuar.

Paso 3: Configure los ajustes de red

La configuración de la red define cómo se implementa el clúster en AWS la infraestructura. Esto incluye la VPC, las zonas y subredes de disponibilidad y los grupos de seguridad que controlan las redes, la disponibilidad y el acceso.

1. Para las redes, haga lo siguiente:

   a. Elija la VPC que desee usar para el clúster.

   b. En función del número de zonas de disponibilidad que haya seleccionado anteriormente, especifique las zonas de disponibilidad y las subredes en las que se desplegarán los agentes.

      Para los corredores estándar de la región de EE. UU. Oeste (norte de California), se necesitan dos subredes en dos zonas de disponibilidad diferentes. En el resto de las regiones donde esté disponible Amazon MSK, puede especificar dos o tres subredes. Todas las subredes deben estar en diferentes zonas de disponibilidad.

      Para los corredores de Express, necesita tres subredes en tres zonas de disponibilidad diferentes.

      Al crear un clúster aprovisionado por MSK, MSK distribuye los nodos del intermediario de manera uniforme en las subredes que especifique.

   c. Para los grupos de seguridad en Amazon EC2, elige o crea uno o más grupos de seguridad a los que quieras dar acceso a tu clúster. Estos grupos EC2 de seguridad de Amazon controlan el tráfico entrante y saliente a sus agentes. Por ejemplo, los grupos de seguridad de las máquinas cliente.

      Si especifica grupos de seguridad que se comparten con usted, debe asegurarse de que tiene permisos para usarlos. En concreto, necesita el permiso `ec2:DescribeSecurityGroups`. Para obtener más información, consulte Conectarse a un clúster de MSK.

2. Elija Paso siguiente para continuar.

Paso 4: Configure los ajustes de seguridad

1. En la sección Configuración de seguridad, haga lo siguiente:

- Elija uno o varios de los siguientes métodos de autenticación y autorización para controlar el acceso de los clientes a sus clústeres de Kafka:

  - Acceso no autenticado: permite a los clientes acceder al clúster sin proporcionar ninguna credencial de autenticación. Este método supone un riesgo para la seguridad y es posible que no cumpla con las prácticas recomendadas de seguridad. Para obtener más información, consulte msk-unrestricted-access-check.
  - Autenticación basada en roles de IAM: permite la autenticación y autorización de los clientes mediante AWS usuarios/roles de IAM. Este método proporciona un control detallado del acceso a los clústeres mediante políticas de IAM. Recomendamos este método para las aplicaciones que ya se están ejecutando. AWS
  - Autenticación SASL/SCRAM: requiere que los clientes proporcionen las credenciales de nombre de usuario y contraseña almacenadas para la autenticación. AWS Secrets Manager Amazon MSK recupera estas credenciales de Secrets Manager y autentica a los usuarios de forma segura.

    Para configurar las credenciales de inicio de sesión relacionadas con la autenticación de un clúster, cree primero un recurso secreto en Secrets Manager. A continuación, asocie las credenciales de inicio de sesión a ese secreto. Para obtener más información sobre este método de control de acceso, consulteConfigurar la SASL/SCRAM autenticación para un clúster de Amazon MSK.
  - Autenticación de clientes mediante TLS AWS Certificate Manager (ACM): permite la autenticación mutua entre clientes y agentes mediante certificados digitales. Debe configurar un AWS Private Certificate Authority (AWS Private CA) igual o diferente al de su Cuenta de AWS clúster.

    Recomendamos encarecidamente utilizar AWS Private CA s independientes para cada clúster de MSK al implementar mTLS. Esto garantiza que los certificados TLS firmados por PCAs solo se autentiquen con un único clúster de MSK, manteniendo así un estricto control de acceso.

2. En Cifrado, elija el tipo de clave KMS que desee usar para cifrar los datos en reposo. Para obtener más información, consulte the section called “Cifrado en reposo de Amazon MSK”.

   El cifrado de los datos en reposo protege la integridad de los datos almacenados, mientras que el cifrado en tránsito protege la confidencialidad de los datos frente a la supervisión de la red durante la transferencia.

3. Elija Paso siguiente para continuar.

Paso 5: Configurar las opciones de supervisión

Este procedimiento describe cómo configurar las métricas de los corredores y cómo recopilar y entregar los registros de los corredores. Con esta configuración, puede observar y analizar el estado y el rendimiento de su clúster y solucionar problemas. Para obtener más información, consulte the section called “Supervisión de un clúster”.

1. Para CloudWatch las métricas de Amazon de este clúster, elige uno de los siguientes niveles de supervisión. Las métricas recopiladas en cada nivel de monitoreo se integran CloudWatch para la visualización y las alertas.

   a. Supervisión básica: proporciona un conjunto de métricas esenciales a nivel de clúster sin coste adicional. Este nivel es adecuado para la mayoría de los casos de uso con necesidades generales de supervisión.

   b. Supervisión mejorada a nivel de corredor: proporciona métricas detalladas de los corredores a un costo adicional. Este nivel incluye la supervisión básica y métricas de los corredores más detalladas, como las métricas de almacenamiento en niveles, los bytes in/out de otros corredores y el tiempo total de las operaciones. read/write Pagas por las métricas de este nivel, mientras que las métricas del nivel básico siguen siendo gratuitas.

   c. Supervisión mejorada a nivel de tema: proporciona métricas para temas individuales a un costo adicional. Elija este nivel para obtener una visión más detallada del rendimiento de los temas en todos los corredores. Este nivel incluye una supervisión mejorada a nivel de agente y métricas a nivel de tema, como las métricas de almacenamiento por niveles para un tema específico y la cantidad de mensajes recibidos por segundo.

   d. Supervisión mejorada a nivel de partición: proporciona la vista más detallada de las métricas por partición con un coste adicional. Elija este nivel para obtener la supervisión más detallada mediante la captura de las métricas de cada partición dentro de cada tema de los distintos corredores. Este nivel incluye una supervisión mejorada a nivel de tema y métricas detalladas específicas de cada partición, como las métricas de desfase de compensación.

   Para obtener más información sobre las métricas disponibles para los tipos de corredores estándar y exprés en cada uno de estos niveles de supervisión, consulte y. CloudWatch métricas para corredores estándar CloudWatch métricas para corredores de Express

2. (Opcional) Si desea exportar métricas en formato Prometheus mediante JMX Exporter, Node Exporter o ambos, seleccione Habilitar la supervisión abierta con Prometheus. Para obtener más información acerca de esta opción, consulta Supervisión con Prometheus.
3. (Opcional) Para configurar su clúster de MSK de forma que envíe los registros de los corredores a varios agentes Servicios de AWS para la solución de problemas y la auditoría, elija una o varias de las siguientes opciones. Amazon MSK no crea estos recursos de destino si aún no existen. Para obtener más información, consulte Registros de agente.

   - Entregar a Amazon CloudWatch Logs: envía los registros a CloudWatch con funciones de agrupamiento, búsqueda y visualización. Puede consultar y analizar los registros sin salir del AWS Management Console.
   - Entregar a Amazon S3: almacena los registros como archivos en depósitos de Amazon S3 para archivarlos a largo plazo y analizarlos por lotes.
   - Entregar a Amazon Data Firehose: envía los registros a Firehose para su entrega automática al OpenSearch Servicio de Amazon para solucionar problemas en tiempo real.
4. (Opcional) Para ayudarte a identificar, organizar o buscar tu clúster, selecciona Añadir nueva etiqueta para añadir etiquetas como pares clave-valor. Por ejemplo, agrega una etiqueta a tu clúster con el par clave-valor de y. **`Load testing`** **`Test`**

   Para obtener más información sobre el uso de etiquetas en los clústeres, consulte. Etiquetado de un clúster de Amazon MSK
5. Elija Paso siguiente para continuar.

Paso 6: Revise la configuración del clúster

1. Revise la configuración del clúster.

   Seleccione Editar o Anterior para cambiar cualquiera de los ajustes que especificó anteriormente o volver a la pantalla de la consola anterior.
2. Elija Create cluster.
3. Compruebe el estado de este clúster en la sección de resumen del clúster de la página de detalles del clúster. El estado cambia de En creación a Activo a medida que Amazon MSK aprovisiona el clúster. Cuando el estado sea Activo, puede conectarse al clúster. Para obtener más información acerca del estado de un clúster, consulte Conozca los estados de los clústeres aprovisionados por MSK.

Cree un clúster de Amazon MSK aprovisionado mediante AWS CLI

1. Copie el siguiente JSON y guárdelo en un archivo. Nombre el archivo `brokernodegroupinfo.json`. Sustituya la subred IDs del JSON por los valores que correspondan a sus subredes. Estas subredes deben estar en diferentes zonas de disponibilidad. *"Security-Group-ID"*Sustitúyalo por el ID de uno o más grupos de seguridad de la VPC del cliente. Los clientes asociados a estos grupos de seguridad obtienen acceso al clúster. Si especifica grupos de seguridad que se han compartido con usted, debe asegurarse de que tiene permisos para ellos. En concreto, necesita el permiso `ec2:DescribeSecurityGroups`. Para ver un ejemplo, consulte Amazon EC2: permite gestionar los grupos de EC2 seguridad de Amazon asociados a una VPC específica, mediante programación y en la consola. Por último, guarde el archivo JSON actualizado en el ordenador en el que lo tiene instalado. AWS CLI

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

> **Important**
>
> Para los corredores de Express, necesita tres subredes en tres zonas de disponibilidad diferentes. Tampoco es necesario definir ninguna propiedad relacionada con el almacenamiento.
> Para los corredores estándar de la región EE.UU. Oeste (Norte de California), se necesitan dos subredes en dos zonas de disponibilidad diferentes. En el resto de las regiones donde esté disponible Amazon MSK, puede especificar dos o tres subredes. Todas las subredes deben estar en diferentes zonas de disponibilidad. Al crear un clúster, Amazon MSK distribuye los nodos del agente de manera uniforme a través de las subredes que indique.

2. Ejecute el siguiente AWS CLI comando en el directorio en el que guardó el `brokernodegroupinfo.json` archivo y *`"Your-Cluster-Name"`* sustitúyalo por el nombre que prefiera. Para*`"Monitoring-Level"`*, puede especificar uno de los tres valores siguientes:`DEFAULT`,`PER_BROKER`, o`PER_TOPIC_PER_BROKER`. Para obtener información sobre estos tres niveles diferentes de supervisión, consulte ???. El parámetro `enhanced-monitoring` es opcional. Si no lo especifica en el comando `create-cluster`, obtendrá el nivel de supervisión `DEFAULT`.

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

El resultado del comando tendrá un aspecto similar al siguiente JSON:

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

> **Note**
>
> El comando `create-cluster` puede devolver un error que indica que una o más subredes pertenecen a zonas de disponibilidad no compatibles. Cuando esto sucede, el error indica qué zonas de disponibilidad no son compatibles. Cree subredes que no utilicen las zonas de disponibilidad no admitidas e intente ejecutar el comando `create-cluster` de nuevo.

3. Guarde el valor de la clave `ClusterArn`, ya que lo necesitará para realizar otras acciones en el clúster.
4. Ejecute el siguiente comando para comprobar el `STATE` del clúster. El valor de `STATE` cambia de `CREATING` a `ACTIVE` a medida que Amazon MSK aprovisiona el clúster. Cuando el estado sea `ACTIVE`, podrá conectarse al clúster. Para obtener más información acerca del estado de un clúster, consulte Conozca los estados de los clústeres aprovisionados por MSK.

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

Cree un clúster aprovisionado de MSK con una configuración de Amazon MSK personalizada mediante el AWS CLI

Para obtener información acerca de las configuraciones de Amazon MSK personalizadas y cómo crearlas, consulte the section called “Configuración de bróker”.

1. Guarde el siguiente JSON en un archivo y *configuration-arn* sustitúyalo por el ARN de la configuración que desee usar para crear el clúster.

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

2. Ejecute el comando `create-cluster` y use la opción `configuration-info` para apuntar al archivo JSON que guardó en el paso anterior. A continuación se muestra un ejemplo.

```
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
```

   El siguiente es un ejemplo de una respuesta correcta después de ejecutar este comando.

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

Cree un clúster aprovisionado de MSK mediante la API de Amazon MSK

La API de Amazon MSK le permite crear y gestionar mediante programación su clúster aprovisionado de MSK como parte de scripts de implementación o aprovisionamiento de infraestructura automatizados.

Para crear un clúster aprovisionado de MSK mediante la API, consulte. CreateCluster

Enumeración de clústeres de Amazon MSK

Para obtener un agente de arranque para un clúster de Amazon MSK, necesita el nombre de recurso de Amazon (ARN) del clúster. Si no tiene el ARN para su clúster, puede encontrarlo enumerando todos los clústeres. Consulte the section called “Obtenga los corredores de bootstrap”.

Temas

- Enumere los clústeres mediante la AWS Management Console
- Enumere los clústeres mediante el AWS CLI
- Enumeración de clústeres mediante la API

Enumere los clústeres mediante la AWS Management Console

Para obtener un agente de arranque para un clúster de Amazon MSK, necesita el nombre de recurso de Amazon (ARN) del clúster. Si no tiene el ARN para su clúster, puede encontrarlo enumerando todos los clústeres. Consulte the section called “Obtenga los corredores de bootstrap”.

1. ¿Iniciar sesión en la AWS Management Console consola Amazon MSK y abrirla desde https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. La tabla muestra todos los clústeres de la región actual en esta cuenta. Elija el nombre de un clúster para ver sus detalles.

Enumere los clústeres mediante el AWS CLI

Para obtener un agente de arranque para un clúster de Amazon MSK, necesita el nombre de recurso de Amazon (ARN) del clúster. Si no tiene el ARN para su clúster, puede encontrarlo enumerando todos los clústeres. Consulte the section called “Obtenga los corredores de bootstrap”.

```
aws kafka list-clusters
```

Enumeración de clústeres mediante la API

Para obtener un agente de arranque para un clúster de Amazon MSK, necesita el nombre de recurso de Amazon (ARN) del clúster. Si no tiene el ARN para su clúster, puede encontrarlo enumerando todos los clústeres. Consulte the section called “Obtenga los corredores de bootstrap”.

Para enumerar los clústeres que utilizan la API, consulte ListClusters.

Conexión a un clúster de Amazon MSK

De forma predeterminada, los clientes pueden acceder a un clúster de MSK aprovisionado solo si están en la misma VPC que el clúster. Todas las comunicaciones entre los clientes de Kafka y su clúster de MSK aprovisioned son privadas de forma predeterminada y los datos de streaming nunca pasan por Internet. Para acceder a su clúster aprovisionado de MSK desde un cliente que esté en la misma VPC que el clúster, asegúrese de que el grupo de seguridad del clúster tenga una regla de entrada que acepte tráfico del grupo de seguridad del cliente. Para obtener información acerca de estas reglas, consulte Reglas del grupo de seguridad. Para obtener un ejemplo de cómo acceder a un clúster desde una EC2 instancia de Amazon que esté en la misma VPC que el clúster, consulte. the section called “Introducción”

> **Note**
>
> KRaft el modo de metadatos y los agentes de MSK Express no pueden habilitar tanto la supervisión abierta como el acceso público.

Para acceder a su clúster de MSK aprovisionado desde un cliente que esté fuera de la VPC del clúster, consulte Access from within AWS but outside cluster's VPC.

Temas

- Activación del acceso público a un clúster de MSK
- Acceso desde dentro de AWS , pero fuera de la VPC del clúster

Activación del acceso público a un clúster de MSK

Amazon MSK le ofrece la opción de activar el acceso público a los agentes de clústeres de MSK que ejecutan la versión 2.6.0 o posterior de Apache Kafka. Por motivos de seguridad, no puede activar el acceso público al crear un clúster de MSK. Sin embargo, puede actualizar un clúster existente para que sea de acceso público. También puede crear un clúster nuevo y, a continuación, actualizarlo para que sea accesible públicamente.

Puede activar el acceso público a un clúster de MSK sin costo adicional, pero se aplican los costos de transferencia de AWS datos de estándar para la transferencia de datos dentro y fuera del clúster. Para obtener información acerca de los precios, consulte Precios de Amazon EC2 On-Demand.

> **Note**
>
> Si utiliza los métodos de control de acceso SASL/SCRAM o mTLS, primero debe configurar Apache Kafka para su clúster. ACLs A continuación, actualiza la configuración del clúster para establecer la propiedad en false. `allow.everyone.if.no.acl.found` Para obtener información acerca de cómo actualizar la configuración de un clúster, consulte the section called “Operaciones de configuración de corredores”.

Para activar el acceso público a un clúster de MSK, asegúrese de que el clúster cumpla todas las condiciones siguientes:

- Las subredes asociadas al clúster deben ser públicas. Cada subred pública tiene una IPv4 dirección pública asociada y el precio de IPv4 las direcciones públicas se indica en la página de precios de Amazon VPC. Esto significa que las subredes deben tener una tabla de enrutamiento asociada con una puerta de enlace de Internet adjunta. Para obtener más información sobre cómo crear y asociar una puerta de enlace de Internet, consulte Concesión del acceso a internet de la VPC con puertas de enlace de Internet en la Guía del usuario de Amazon VPC.
- El control de acceso no autenticado debe estar activado y al menos uno de los siguientes métodos de control de acceso debe estar activado:, mTLS. SASL/IAM, SASL/SCRAM Para obtener más información acerca de cómo actualizar el método de control de acceso de un clúster, consulte the section called “Actualice la seguridad del clúster”.
- El cifrado dentro del clúster debe estar activado. Esta configuración es la predeterminada al crear un clúster. No es posible activar el cifrado dentro del clúster para un clúster que se creó con el cifrado desactivado. Por lo tanto, no es posible activar el acceso público a un clúster que se creó con el cifrado dentro del clúster desactivado.
- El tráfico de texto sin formato entre los agentes y los clientes debe estar desactivado. Para obtener información sobre cómo desactivarlo si está activado, consulte the section called “Actualice la seguridad del clúster”.
- Si utiliza el control de acceso de IAM y desea aplicar políticas de autorización o actualizarlas, consultethe section called “Control de acceso de IAM”. Para obtener información sobre Apache Kafka ACLs, consultethe section called “Apache Kafka ACLs”.

Una vez que se asegure de que un clúster de MSK cumple las condiciones enumeradas anteriormente, puede usar la AWS Management Console AWS CLI, la o la API de Amazon MSK para activar el acceso público. Después de activar el acceso público a un clúster, puede obtener

una cadena pública de bootstrap-brokers para este. Para obtener más información acerca de cómo obtener los agentes de arranque de un clúster, consulte the section called “Obtenga los corredores de bootstrap”.

> **Important**
>
> Además de activar el acceso público, asegúrese de que los grupos de seguridad del clúster tengan reglas de TCP de entrada que permitan el acceso público desde su dirección IP. Le recomendamos que estas reglas sean lo más restrictivas posible. Para obtener más información acerca de los grupos de seguridad y las reglas de entrada, consulte Grupos de seguridad de la VPC en la Guía del usuario de Amazon VPC. Para ver los números de los puertos, consulte the section called “Información del puerto”. Para obtener instrucciones acerca de cómo cambiar el grupo de seguridad de un clúster, consulte the section called “Modificación de los grupos de seguridad”.

> **Note**
>
> Si sigue estas instrucciones para activar el acceso público y, a pesar de ello, sigue sin poder acceder al clúster, consulte the section called “No se puede acceder al clúster que tiene activado el acceso público”.

Activación del acceso público mediante la consola

1. ¿Iniciar sesión en la AWS Management Console y abrir la consola de Amazon MSK en https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. En la lista de clústeres, elija el clúster en el que desea activar el acceso público.
3. Seleccione la pestaña Propiedades y, a continuación, busque la sección Configuración de redes.
4. Elija Editar el acceso público.

Activación del acceso público mediante la AWS CLI

1. Antes de ejecutar el siguiente AWS CLI comando de la, reemplace *`ClusterArn`* y *`Current-Cluster-Version`* por el ARN y la versión actual del clúster. Para encontrar la versión actual del clúster, utilice la DescribeClusteroperación o el comando AWS CLI describe-cluster. Un ejemplo de ID de versión es `KTVPDKIKX0DER`.

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

El resultado de este comando `update-connectivity` tendrá un aspecto similar al siguiente.

```
{
    "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**
>
> Para desactivar el acceso público, use un AWS CLI comando de la similar, pero con la siguiente información de conectividad en su lugar:
>
> ```
> '{"PublicAccess": {"Type": "DISABLED"}}'
> ```

2. Para obtener el resultado de la `update-connectivity` operación, ejecute el siguiente comando, debe *`ClusterOperationArn`* sustituir por el ARN que obtuvo en la salida del `update-connectivity` comando.

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

El resultado de este comando `describe-cluster-operation` tendrá un aspecto similar al siguiente.

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

Si `OperationState` tiene el valor `UPDATE_IN_PROGRESS`, espere un rato y vuelva a ejecutar el comando `describe-cluster-operation`.

Activación del acceso público mediante la API de Amazon MSK

- Para usar la API para activar o desactivar el acceso público a un clúster, consulte UpdateConnectivity.

> **Note**
>
> Por motivos de seguridad, Amazon MSK no permite el acceso público a Apache ZooKeeper o a los nodos de KRaft controlador.

Acceso desde dentro de AWS , pero fuera de la VPC del clúster

Para conectarse a un clúster de MSK desde dentro de AWS , pero fuera de la VPC de Amazon del clúster, existen las siguientes opciones.

Interconexión de Amazon VPC

Para acceder a su clúster de MSK desde una VPC diferente de la VPC del clúster, puede crear un emparejamiento entre las dos. VPCs Para obtener información sobre las interconexiones de VPC, consulte la Guía de interconexión de Amazon VPC.

AWS Direct Connect

AWS Direct Connect vincula su red en las instalaciones con a AWS través de un cable estándar Ethernet de fibra óptica de 1 o 10 Gigabits. Un extremo del cable está conectado al router y el otro a un router. AWS Direct Connect Con esta conexión implementada, puede crear interfaces virtuales directamente en la AWS nube de y la VPC de Amazon omitiendo a los proveedores de servicios de Internet en su ruta de acceso a la red. Para obtener más información, consulte AWS Direct Connect.

AWS Transit Gateway

AWS Transit Gateway es un servicio que le permite conectar sus redes locales VPCs y las suyas a una única puerta de enlace. Para obtener información acerca de cómo utilizar AWS Transit Gateway, consulte AWS Transit Gateway.

Conexiones de VPN

Puede conectar la VPC de su clúster de MSK a redes y usuarios remotos mediante las opciones de conectividad de VPN descritas en el tema siguiente: Conexiones de VPN.

Proxies REST

Puede instalar una proxy REST en una instancia en ejecución en la VPC de Amazon del clúster. Los proxies REST permiten a los productores y los consumidores comunicarse con el clúster mediante solicitudes de API HTTP.

Conectividad con varias VPC en regiones diferentes

En el siguiente documento se describen las opciones de conectividad con varias VPCs regiones: Multiple Region Multi-VPC Connectivity.

Conectividad privada con varias VPC en una sola región

La conectividad privada con varias VPC (con tecnología de AWS PrivateLink) para los clústeres de Amazon Managed Streaming para Apache Kafka (Amazon MSK) es una característica que le permite conectar más rápidamente los clientes de Kafka alojados en diferentes VPCs nubes privadas virtuales () AWS y cuentas de a un clúster de Amazon MSK.

Consulte Single Region multi-VPC connectivity for cross-account clients.

EC2-Fin de la compatibilidad con la red clásica

Amazon MSK ya no es compatible con las EC2 instancias de Amazon que se ejecutan en la red Amazon EC2 Classic.

Consulte EC2-Classic Networking is Retiring — Here's How to Prepare.

Conectividad privada con varias VPC de Amazon MSK en una sola región

La conectividad privada con varias VPC (con tecnología de AWS PrivateLink) para los clústeres de Amazon Managed Streaming para Apache Kafka (Amazon MSK) es una característica que le permite conectar más rápidamente los clientes de Kafka alojados en diferentes VPCs nubes privadas virtuales () AWS y cuentas de a un clúster de Amazon MSK.

La conectividad privada con varias VPC es una solución administrada que simplifica la infraestructura de red para la conectividad con varias VPC y entre cuentas. Los clientes pueden conectarse al clúster de Amazon MSK y, PrivateLink al mismo tiempo, mantener todo el tráfico dentro de la AWS red. La conectividad privada con varias VPC para los clústeres de Amazon MSK está disponible en todas las regiones de en las AWS que Amazon MSK está disponible.

Temas

- ¿Qué es la conectividad privada con varias VPC?
- Ventajas de la conectividad privada con varias VPC
- Requisitos y limitaciones de la conectividad privada con varias VPC
- Introducción a la conectividad privada con varias VPC
- Actualización de los esquemas de autorización de un clúster
- Rechazo de una conexión de VPC administrada a un clúster de Amazon MSK
- Eliminación de una conexión de VPC administrada a un clúster de Amazon MSK
- Permisos para la conectividad privada con varias VPC

¿Qué es la conectividad privada con varias VPC?

La conectividad privada con varias VPC para Amazon MSK es una opción de conectividad que le permite conectar clientes de Apache Kafka que están alojados en diferentes nubes privadas virtuales (VPCs) y AWS cuentas de a un clúster de MSK.

Amazon MSK simplifica el acceso entre cuentas con políticas de clústeres. Estas políticas permiten al propietario del clúster conceder permisos a otras AWS cuentas de para establecer una conectividad privada con el clúster de MSK.

Ventajas de la conectividad privada con varias VPC

La conectividad privada con varias VPC tiene varias ventajas en comparación con otras soluciones de conectividad:

- Automatiza la administración operativa de la solución de AWS PrivateLink conectividad.
- Permite superponer conexiones IPs VPCs, lo que elimina la necesidad de mantener tablas de interconexiones complejas y no superpuestas IPs y de enrutamiento asociadas a otras soluciones de conectividad de VPC.

Utiliza una política de clústeres para su clúster de MSK a fin de definir qué AWS cuentas de tienen permisos para configurar la conectividad privada entre cuentas con su clúster de MSK. El administrador de varias cuentas puede delegar los permisos a los roles o usuarios adecuados. Si se utiliza con la autenticación de clientes de IAM, también puede utilizar la política de clústeres para definir los permisos del plano de datos de Kafka de forma granular para los clientes que se conectan.

Requisitos y limitaciones de la conectividad privada con varias VPC

Tenga en cuenta estos requisitos del clúster de MSK para ejecutar la conectividad privada con varias VPC:

- La conectividad privada con varias VPC solo se admite en la versión 2.7.1 o superior de Apache Kafka. Asegúrese de que todos los clientes que utilice con el clúster de MSK ejecuten versiones de Apache Kafka que sean compatibles con el clúster.
- La conectividad privada con varias VPC admite los tipos de autenticación IAM, TLS y SASL/SCRAM. Los clústeres no autenticados no pueden utilizar la conectividad privada con varias VPC.
- Si utiliza los métodos de control de acceso SASL/SCRAM o mTLS, debe configurar Apache Kafka para su clúster. ACLs En primer lugar, configure Apache Kafka ACLs para su clúster. A continuación, actualice la configuración del clúster para que la propiedad `allow.everyone.if.no.acl.found` del clúster esté establecida en false. Para obtener información acerca de cómo actualizar la configuración de un clúster, consulte the section called “Operaciones de configuración de corredores”. Si utiliza el control de acceso de IAM y desea aplicar políticas de autorización o actualizarlas, consulte the section called “Control de acceso de

IAM". Para obtener información sobre Apache Kafka ACLs, consultethe section called "Apache Kafka ACLs".

- La conectividad privada con varias VPC no admite el tipo de instancia t3.small.
- La conectividad privada con varias VPC no se admite en todas AWS las regiones de, solo en las AWS cuentas de de la misma región.
- Para configurar la conectividad privada de varias VPC, debe tener el mismo número de subredes de cliente que de subredes de clúster. También debe asegurarse de que la zona de disponibilidad sea la misma para la IDs subred del cliente y la subred del clúster.
- Amazon MSK no admite la conectividad privada con varias VPC con los nodos de ZooKeeper.

Introducción a la conectividad privada con varias VPC

Temas

- Paso 1: en el clúster de MSK de la cuenta A, active la conectividad con varias VPC para el esquema de autenticación de IAM en el clúster
- Paso 2: asociación de una política de clúster al clúster de MSK
- Paso 3: acciones del usuario entre cuentas para configurar las conexiones de VPC administradas por el cliente

En este tutorial, se utiliza un caso de uso común como ejemplo de cómo se puede utilizar la conectividad con varias VPC para conectar de forma privada un cliente de Apache Kafka a un clúster de MSK desde dentro de, AWS pero desde fuera de la VPC del clúster. Este proceso requiere que el usuario entre cuentas cree una conexión y una configuración de VPC administrada por MSK para cada cliente, incluidos los permisos de cliente necesarios. El proceso también requiere que el propietario del clúster de MSK habilite la PrivateLink conectividad en el clúster de MSK y seleccione esquemas de autenticación para controlar el acceso al clúster.

En diferentes partes de este tutorial, seleccionamos las opciones que se aplican a este ejemplo. Esto no significa que sean las únicas opciones que funcionan para configurar un clúster de MSK o instancias de cliente.

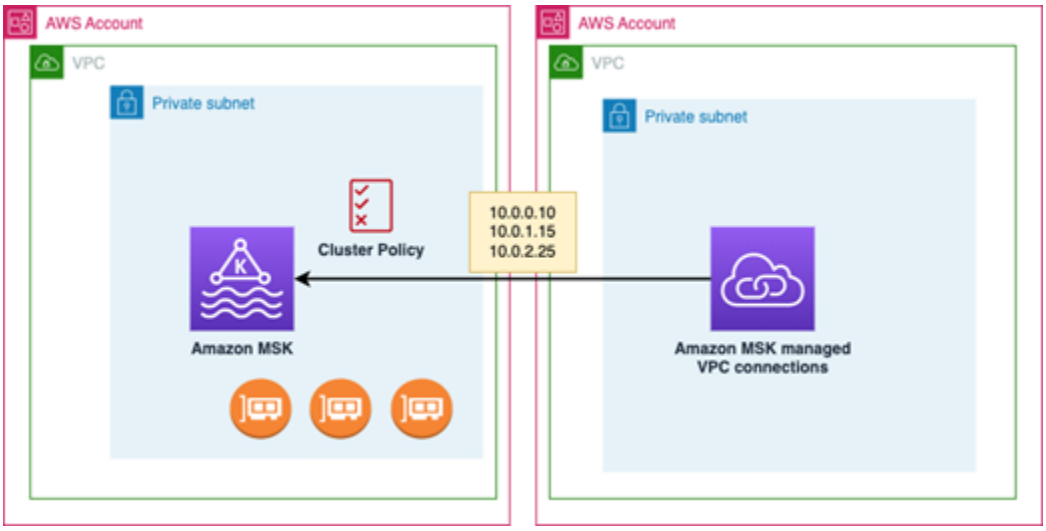
La configuración de red para este caso de uso es la siguiente:

- Un usuario entre cuentas (cliente de Kafka) y un clúster de MSK están en la misma red o región de AWS , pero en cuentas diferentes:
  - Clúster de MSK en la cuenta A

  - Cliente de Kafka en la cuenta B
- El usuario entre cuentas se conectará de forma privada al clúster de MSK mediante el esquema de autenticación de IAM.

En este tutorial se presupone que hay un clúster de MSK aprovisionado creado con la versión 2.7.1 o superior de Apache Kafka. El clúster de MSK debe estar en estado ACTIVO antes de comenzar el proceso de configuración. Para evitar posibles pérdidas de datos o tiempos de inactividad, los clientes que usen una conexión privada con varias VPC para conectarse al clúster deben usar versiones de Apache Kafka que sean compatibles con el clúster.

El siguiente diagrama ilustra la arquitectura de la conectividad de varias cuentas de Amazon MSK conectada a un cliente de una cuenta de diferente. AWS



Paso 1: en el clúster de MSK de la cuenta A, active la conectividad con varias VPC para el esquema de autenticación de IAM en el clúster

El propietario del clúster de MSK debe realizar los ajustes de configuración en el clúster de MSK una vez creado el clúster y ponerlo en estado ACTIVO.

El propietario del clúster activa la conectividad privada con varias VPC en el clúster ACTIVO para cualquier esquema de autenticación que esté activo en el clúster. Esto se puede hacer mediante la UpdateSecurity API o la consola de MSK. Los esquemas de autenticación IAM, SASL/SCRAM y TLS admiten la conectividad privada con varias VPC. La conectividad privada con varias VPC no se puede habilitar para los clústeres no autenticados.

En este caso de uso, configurará el clúster para que utilice el esquema de autenticación de IAM.

> **Note**
>
> Si va a configurar su clúster de MSK para usar el esquema de autenticación SASL/SCRAM, la propiedad de Apache Kafka ACLs «» es obligatoria. `allow.everyone.if.no.acl.found=false` ACLsConsulte Apache Kafka.

Al actualizar la configuración de conectividad privada con varias VPC, Amazon MSK inicia un reinicio continuo de los nodos del agente para actualizar las configuraciones del agente. Esta acción puede tardar hasta 30 minutos o más en completarse. No puede realizar otras actualizaciones en el clúster mientras se actualiza la conectividad.

Active la opción con varias VPC para los esquemas de autenticación seleccionados en el clúster de la cuenta A mediante la consola

1. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/la cuenta en la que se encuentre el clúster.
2. En el panel de navegación, en Clústeres de MSK, seleccione Clústeres para ver la lista de clústeres de la cuenta.
3. Seleccione el clúster que desee configurar para la conectividad privada con varias VPC. El clúster debe estar en estado ACTIVO.
4. Seleccione la pestaña Propiedades del clúster y, a continuación, vaya a Configuración de redes.
5. Seleccione el menú desplegable Editar y elija Activar la conectividad con varias VPC.
6. Seleccione uno o más tipos de autenticación que desee activar para este clúster. Para este caso de uso, seleccione la autenticación basada en roles de IAM.
7. Seleccione Guardar cambios.

Example : UpdateConnectivity API que activa los esquemas de autenticación de conectividad privada con varias VPC en un clúster

Como alternativa a la consola de MSK, puede usar la UpdateConnectivity API para activar la conectividad privada con varias VPC y configurar esquemas de autenticación en un clúster ACTIVO. En el siguiente ejemplo, se muestra el esquema de autenticación de IAM para el clúster.

```
{
  "currentVersion": "K3T4TT2Z381HKD",
```

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

Amazon MSK crea la infraestructura de red necesaria para la conectividad privada. Amazon MSK también crea un nuevo conjunto de puntos de conexión de agente de arranque para cada tipo de autenticación que requiera conectividad privada. Tenga en cuenta que el esquema de autenticación de texto sin formato no admite la conectividad privada con varias VPC.

Paso 2: asociación de una política de clúster al clúster de MSK

El propietario del clúster puede asociar una política de clúster (también conocida como política basada en recursos) al clúster de MSK, donde activará la conectividad privada con varias VPC. La política de clúster otorga a los clientes permiso para acceder al clúster desde otra cuenta. Para poder editar la política de clúster, necesitará los ID de las cuentas que deben tener permiso para acceder al clúster de MSK. Consulte How Amazon MSK works with IAM.

El propietario del clúster debe asociar una política de clúster al clúster de MSK que autorice al usuario con varias cuentas de la cuenta B a contratar agentes de arranque para el clúster y a autorizar las siguientes acciones en el clúster de MSK de la cuenta A:

- CreateVpcConnection
- GetBootstrapBrokers
- DescribeCluster
- DescribeClusterV2

Example

Como referencia, a continuación se muestra un ejemplo del JSON para una política de clúster básica, similar a la política predeterminada que se muestra en el editor de políticas de IAM de la

consola de MSK. La siguiente política concede permisos para el acceso a nivel de clúster, tema y grupo.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "123456789012"
      },
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:GetBootstrapBrokers",
        "kafka:DescribeCluster",
        "kafka:DescribeClusterV2",
        "kafka-cluster:*"
      ],
      "Resource": "arn:aws:kafka:us-east-1:111122223333:cluster/testing/
de8982fa-8222-4e87-8b20-9bf3cdfa1521-2"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "123456789012"
      },
      "Action": "kafka-cluster:*",
      "Resource": "arn:aws:kafka:us-east-1:111122223333:topic/testing/*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "123456789012"
      },
      "Action": "kafka-cluster:*",
      "Resource": "arn:aws:kafka:us-east-1:111122223333:group/testing/*"
    }
  ]
}
```

Asociación de una política de clúster al clúster de MSK

1. En la consola de Amazon MSK, en Clústeres de MSK, elija Clústeres.

2. Desplázate hacia abajo hasta llegar a Configuración de seguridad y seleccione Editar política del clúster.
3. En la consola, en la pantalla Editar política de clúster, seleccione Política básica para la conectividad con varias VPC.
4. En el campo ID de cuenta, ingrese el ID de cuenta de cada cuenta que deba tener permiso para acceder a este clúster. A medida que escriba el ID, este se copia automáticamente en la sintaxis JSON de la política que se muestra. En nuestro ejemplo de política de clúster, el ID de cuenta es `123456789012`.
5. Seleccione Guardar cambios.

Para obtener información sobre la política de clústeres APIs, consulte Amazon MSK resource-based policies.

Paso 3: acciones del usuario entre cuentas para configurar las conexiones de VPC administradas por el cliente

Para configurar la conectividad privada con varias VPC entre un cliente de una cuenta diferente del clúster de MSK, el usuario entre cuentas crea una conexión de VPC administrada para el cliente. Se pueden conectar varios clientes al clúster de MSK repitiendo este procedimiento. Para los fines de este caso de uso, tendrá que configurar un solo cliente.

Los clientes pueden usar los esquemas de autenticación compatibles: IAM, SASL/SCRAM o TLS. Cada conexión de VPC administrada solo puede tener asociado un esquema de autenticación. El esquema de autenticación del cliente debe configurarse en el clúster de MSK al que se conectará el cliente.

Para este caso de uso, configure el esquema de autenticación del cliente de modo que el cliente de la cuenta B utilice el esquema de autenticación de IAM.

Requisitos previos

Este proceso requiere los siguientes elementos:

- La política de clústeres creada anteriormente que concede al cliente de la cuenta B permiso para realizar acciones en el clúster de MSK de la cuenta A.
- Una política de identidad asociada al cliente en la cuenta B que otorga permisos para `kafka:CreateVpcConnection`, `ec2:CreateTags`, `ec2:CreateVPCEndpoint` y la acción `ec2:DescribeVpcAttribute`.

Example

Como referencia, a continuación se muestra un ejemplo de JSON para una política de identidad de cliente básica.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kafka:CreateVpcConnection",
        "ec2:CreateTags",
        "ec2:CreateVPCEndpoint",
        "ec2:DescribeVpcAttribute"
      ],
      "Resource": "*"
    }
  ]
}
```

Creación de una conexión de VPC administrada para un cliente de la cuenta B

1. Del administrador del clúster, obtenga el ARN del clúster de MSK de la cuenta A al que desea que se conecte el cliente de la cuenta B. Anote el ARN del clúster para usarlo más adelante.
2. En la consola de MSK de la cuenta de cliente B, seleccione Conexiones de VPC administradas y, a continuación, seleccione Crear conexión.
3. En el panel Configuración de conexión, pegue el ARN del clúster en el campo de texto ARN del clúster y, a continuación, seleccione Verificar.
4. Seleccione el Tipo de autenticación para el cliente en la cuenta B. Para este caso de uso, elija IAM al crear la conexión de VPC del cliente.
5. Elija la VPC para el cliente.
6. Elija al menos dos zonas de disponibilidad y subredes asociadas. Puede obtener la zona IDs de disponibilidad desde los detalles del clúster de AWS Management Console o mediante la DescribeClusterAPI o el comando AWS CLI describe-cluster. La zona IDs que especifique para la subred del cliente debe coincidir con las de la subred del clúster. Si faltan los valores de una subred, primero cree una subred con el mismo ID de zona que su clúster de MSK.

7. Elija un Grupo de seguridad para esta conexión de VPC. Puede aceptar el grupo de seguridad predeterminado. Para obtener más información sobre la configuración de grupos de seguridad, consulte Control del tráfico hacia los recursos mediante grupos de seguridad.
8. Seleccione Crear conexión.
9. Para obtener la lista de nuevas cadenas de agente de arranque desde la consola de MSK del usuario entre cuentas (Detalles del clúster > Conexiones de VPC administradas), consulte las cadenas de agente de arranque que se muestran en Cadena de conexión del clúster. Desde la cuenta B del cliente, se puede ver la lista de agentes de arranque llamando a la GetBootstrapBrokersAPI o consultando la lista de agentes de arranque en los detalles del clúster de la consola.
10. Actualice los grupos de seguridad asociados a las conexiones de VPC de la siguiente manera:

    a. Establezca reglas de entrada para la PrivateLink VPC a fin de permitir todo el tráfico del rango de IP desde la red de la cuenta B.

    b. [Opcional] Establezca conectividad mediante Reglas de salida con el clúster de MSK. Elija el Grupo de seguridad en la consola de la VPC, Editar reglas de salida y agregue una regla para Tráfico TCP personalizado para los rangos de puertos del 14001 al 14100. El equilibrador de carga de red de varias VPC escucha en los rangos de puertos del 14001 al 14100. Consulte Network Load Balancers.
11. Configure el cliente de la cuenta B para que utilice los nuevos agentes de arranque para la conectividad privada con varias VPC a fin de conectarse al clúster de MSK de la cuenta A. Consulte Produce and consume data.

Una vez completada la autorización, Amazon MSK crea una conexión de VPC administrada para cada VPC y esquema de autenticación especificados. El grupo de seguridad elegido se asocia a cada conexión. Amazon MSK configura esta conexión de VPC administrada para conectarse de forma privada a los agentes. Puede utilizar el nuevo conjunto de agentes de arrange para conectarse de forma privada al clúster de Amazon MSK.

Actualización de los esquemas de autorización de un clúster

La conectividad privada con varias VPC admite varios esquemas de autorización: conectividad SASL/SCRAM, IAM, and TLS. The cluster owner can turn on/off privada para uno o más esquemas de autenticación. El clúster debe estar en estado ACTIVO para realizar esta acción.

Activación de un esquema de autenticación mediante la consola de Amazon MSK

1. Abra la consola de Amazon MSK en AWS Management Console para el clúster que desee editar.
2. En el panel de navegación, en Clústeres de MSK, seleccione Clústeres para ver la lista de clústeres de la cuenta.
3. Seleccione el clúster que desee editar. El clúster debe estar en estado ACTIVO.
4. Seleccione la pestaña Propiedades del clúster y, a continuación, vaya a Configuración de redes.
5. Seleccione el menú desplegable Editar y elija Activar la conectividad con varias VPC para activar un nuevo esquema de autenticación.
6. Seleccione uno o más tipos de autenticación que desee activar para este clúster.
7. Seleccione Activar la selección.

Al activar un nuevo esquema de autenticación, también debe crear nuevas conexiones de VPC administradas para el nuevo esquema de autenticación y actualizar los clientes para que usen los agentes de arranque específicos del nuevo esquema de autenticación.

Desactivación de un esquema de autenticación mediante la consola de Amazon MSK

> **Note**
>
> Al desactivar la conectividad privada con varias VPC para los esquemas de autenticación, se elimina toda la infraestructura relacionada con la conectividad, incluidas las conexiones de VPC administradas.

Al desactivar la conectividad privada con varias VPC para los esquemas de autenticación, las conexiones de VPC existentes del cliente cambian a INACTIVAS y la infraestructura de Privatelink del clúster, incluidas las conexiones de VPC administradas, se elimina del clúster. El usuario entre cuentas solo puede eliminar la conexión de VPC inactiva. Si la conectividad privada se vuelve a activar en el clúster, el usuario entre cuentas debe crear una nueva conexión con el clúster.

1. Abra la consola de Amazon MSK en AWS Management Console.
2. En el panel de navegación, en Clústeres de MSK, seleccione Clústeres para ver la lista de clústeres de la cuenta.
3. Seleccione el clúster que desee editar. El clúster debe estar en estado ACTIVO.

4. Seleccione la pestaña Propiedades del clúster y, a continuación, vaya a Configuración de redes.
5. Seleccione el menú desplegable Editar y elija Desactivar la conectividad con varias VPC (para desactivar un esquema de autenticación).
6. Seleccione uno o más tipos de autenticación que desee desactivar para este clúster.
7. Seleccione Desactivar la selección.

Example Activación o desactivación de un esquema de autenticación con la API

Como alternativa a la consola de MSK, puede usar la UpdateConnectivity API para activar la conectividad privada con varias VPC y configurar esquemas de autenticación en un clúster ACTIVO. En el siguiente ejemplo, se muestran los esquemas de autenticación SASL/SCRAM e IAM activados para el clúster.

Al activar un nuevo esquema de autenticación, también debe crear nuevas conexiones de VPC administradas para el nuevo esquema de autenticación y actualizar los clientes para que usen los agentes de arranque específicos del nuevo esquema de autenticación.

Al desactivar la conectividad privada con varias VPC para los esquemas de autenticación, las conexiones de VPC existentes del cliente cambian a INACTIVAS y la infraestructura de Privatelink del clúster, incluidas las conexiones de VPC administradas, se elimina. El usuario entre cuentas solo puede eliminar la conexión de VPC inactiva. Si la conectividad privada se vuelve a activar en el clúster, el usuario entre cuentas debe crear una nueva conexión con el clúster.

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

```
        "tls": {
          "enabled": FALSE
        }
      }
    }
  }
}

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

Rechazo de una conexión de VPC administrada a un clúster de Amazon MSK

Desde la consola de Amazon MSK de la cuenta de administrador del clúster, puede rechazar una conexión de VPC del cliente. La conexión de VPC del cliente debe estar en el estado DISPONIBLE para poder rechazarse. Es posible que quiera rechazar una conexión de VPC administrada de un cliente que ya no esté autorizado a conectarse a su clúster. Para evitar que las nuevas conexiones de VPC administradas se conecten a un cliente, deniegue el acceso al cliente en la política de clúster. Una conexión rechazada sigue incurriendo en costos hasta que el propietario de la conexión la elimine. Consulte Delete a managed VPC connection to an Amazon MSK cluster.

Rechazo de una conexión de VPC de cliente mediante la consola de MSK

1. Abra la consola de Amazon MSK en AWS Management Console.
2. En el panel de navegación, seleccione Clústeres y desplácese hasta llegar a la lista Configuración de redes > Conexiones de VPC cliente.
3. Seleccione la conexión que desee rechazar y seleccione Rechazar la conexión de la VPC cliente.
4. Confirme que desea rechazar la conexión de VPC del cliente seleccionada.

Para rechazar una conexión de VPC administrada mediante la API, utilice la API `RejectClientVpcConnection`.

Eliminación de una conexión de VPC administrada a un clúster de Amazon MSK

El usuario entre cuentas puede eliminar una conexión de VPC administrada para un clúster de MSK desde la consola de la cuenta del cliente. Como el usuario propietario del clúster no es propietario de

la conexión de VPC administrada, la conexión no se puede eliminar de la cuenta de administrador del clúster. Una vez que se elimina una conexión de VPC, esta deja de incurrir en costos.

Eliminación de una conexión de VPC administrada con la consola de MSK

1. Desde la cuenta del cliente, abra la consola de Amazon MSK en AWS Management Console.
2. En el panel de navegación, seleccione Conexiones de VPC administradas.
3. En la lista de conexiones, seleccione la conexión que quiera eliminar.
4. Confirme que desea eliminar la conexión de VPC.

Para eliminar una conexión de VPC administrada mediante la API, utilice la API `DeleteVpcConnection`.

Permisos para la conectividad privada con varias VPC

En esta sección, se resumen los permisos necesarios para los clientes y los clústeres que utilizan la característica de conectividad privada con varias VPC. La conectividad privada con varias VPC requiere que el administrador del cliente cree permisos en cada cliente que tendrá una conexión de VPC administrada al clúster de MSK. También requiere que el administrador del clúster de MSK habilite la PrivateLink conectividad en el clúster de MSK y seleccione esquemas de autenticación para controlar el acceso al clúster.

Tipo de autenticación del clúster y permisos de acceso a los temas

Active la característica de conectividad privada con varias VPC para los esquemas de autenticación que están habilitados para su clúster de MSK. Consulte Requisitos y limitaciones de la conectividad privada con varias VPC. Si va a configurar su clúster de MSK para usar el esquema de autenticación SASL/SCRAM, la propiedad de Apache Kafka es obligatoria. ACLs `allow.everyone.if.no.acl.found=false` Después de configurar las Apache Kafka ACLs para el clúster, actualice la configuración del clúster para que la propiedad `allow.everyone.if.no.acl.found` se establezca en false para el clúster. Para obtener información acerca de cómo actualizar la configuración de un clúster, consulte Operaciones de configuración del bróker.

Permisos de políticas de clúster entre cuentas

Si un cliente de Kafka está en una AWS cuenta de diferente a la del clúster de MSK, asocie al clúster de MSK una política basada en clústeres que autorice al usuario raíz del cliente a conectarse entre cuentas. Puede editar la política de clúster con varias VPC mediante el editor de políticas de IAM

de la consola de MSK (Configuración de seguridad del clúster > Editar política del clúster) o usar lo siguiente APIs para administrar la política de clúster:

PutClusterPolicy

Asocia la política de clúster al clúster. Puede usar esta API para crear o actualizar la política de clúster de MSK especificada. Si va a actualizar la política, el campo currentVersion es obligatorio en la carga de la solicitud.

GetClusterPolicy

Recupera el texto JSON del documento de política de clúster asociado al clúster.

DeleteClusterPolicy

Elimina la política de clúster.

El siguiente es un ejemplo del JSON para una política de clúster básica, similar al que se muestra en el editor de políticas de IAM de la consola de MSK. La siguiente política concede permisos para el acceso a nivel de clúster, tema y grupo.

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

```
    }]
}
```

Permisos de cliente para la conectividad privada con varias VPC a un clúster de MSK

Para configurar la conectividad privada con varias VPC entre un cliente de Kafka y un clúster de MSK, el cliente necesita una política de identidad asociada que conceda permisos para las acciones `kafka:CreateVpcConnection`, `ec2:CreateTags` y `ec2:CreateVPCEndpoint` al cliente. Como referencia, a continuación se muestra un ejemplo de JSON para una política de identidad de cliente básica.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kafka:CreateVpcConnection",
        "ec2:CreateTags",
        "ec2:CreateVPCEndpoint"
      ],
      "Resource": "*"
    }
  ]
}
```

Información del puerto

Utilice los siguientes números de puerto para que Amazon MSK pueda comunicarse con los equipos cliente:

- Para comunicarse con productores y consumidores con texto sin formato, los agentes utilizan el puerto 9092.
- Para comunicarse con los agentes mediante el cifrado TLS, utilice el puerto 9094 para acceder desde dentro de AWS y el puerto 9194 para el acceso público.
- Para comunicarse con los agentes mediante SASL/SCRAM, utilice el puerto 9096 para acceder desde dentro de AWS y el puerto 9196 para el acceso público.
- Para comunicarse con los agentes de un clúster configurado para usarthe section called “Control de acceso de IAM”, utilice el puerto 9098 para acceder desde dentro de AWS y el puerto 9198 para el acceso público.

- Para comunicarse con Apache ZooKeeper mediante el cifrado TLS, utilice el puerto 2182. ZooKeeper Los nodos de Apache utilizan el puerto 2181 de forma predeterminada.

Obtención de agentes de arranque para un clúster de Amazon MSK

El término agentes de arranque hace referencia a una lista de agentes que un cliente de Apache Kafka puede utilizar para conectarse a un clúster de Amazon MSK. Es posible que esta lista no incluya todos los agentes del clúster. Puede obtener corredores de bootstrap mediante la API AWS Management Console AWS CLI, o Amazon MSK.

Temas

- Obtenga los corredores de bootstrap utilizando la AWS Management Console
- Obtenga los corredores de bootstrap utilizando el AWS CLI
- Obtención de los agentes de arranque mediante la API

Obtenga los corredores de bootstrap utilizando la AWS Management Console

Este proceso describe cómo obtener corredores de arranque para un clúster utilizando el. AWS Management Console El término agentes de arranque hace referencia a una lista de agentes que un cliente de Apache Kafka puede utilizar como punto de partida para conectarse al clúster. Esta lista no incluye necesariamente todos los agentes de un clúster.

1. ¿Iniciar sesión en la AWS Management Console consola Amazon MSK y abrirla desde https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. La tabla muestra todos los clústeres de la región actual en esta cuenta. Elija el nombre de un clúster para ver su descripción.
3. En la página Resumen del clúster, elija Ver información del cliente. Aquí se muestran los intermediarios de arranque, así como la cadena de conexión de Apache. ZooKeeper

Obtenga los corredores de bootstrap utilizando el AWS CLI

Ejecute el siguiente comando, reemplazando *ClusterArn* por el nombre de recurso de Amazon (ARN) que obtuvo al crear el clúster. Si no tiene el ARN para su clúster, puede encontrarlo enumerando todos los clústeres. Para obtener más información, consulte the section called "Enumerar clústeres".

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

En el caso de un clúster de MSK que utiliza the section called “Control de acceso de IAM”, el resultado de este comando tendrá un aspecto similar al siguiente.

```
{
    "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"
}
```

En el siguiente ejemplo, se muestran los agentes de arranque de un clúster con acceso público activado. Usa el `BootstrapBrokerStringPublicSaslIam` para el acceso público y la `BootstrapBrokerStringSaslIam` cadena para el acceso desde dentro AWS.

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

La cadena de agentes de arranque debe contener tres agentes de todas las zonas de disponibilidad en las que está implementado el clúster de MSK (a menos que solo haya dos agentes disponibles).

Obtención de los agentes de arranque mediante la API

Para hacer que los corredores de bootstrap usen la API, consulte GetBootstrapBrokers.

Supervise un clúster aprovisionado de Amazon MSK

Amazon MSK le ayuda a supervisar el estado de su clúster aprovisionado de Amazon MSK de varias maneras.

- Amazon MSK recopila las métricas de Apache Kafka y las envía a Amazon, CloudWatch donde puede verlas. Para obtener más información acerca de las métricas de Apache Kafka, incluidas

las que aparecen en las superficies de Amazon MSK, consulte Monitoring en la documentación de Apache Kafka.

- También puede supervisar su clúster de MSK con Prometheus, una aplicación de supervisión de código abierto. Para obtener información acerca de Prometheus, consulte Overview en la documentación de Prometheus. Para obtener información sobre cómo supervisar su clúster aprovisionado de MSK con Prometheus, consulte. the section called “Supervisión con Prometheus”
- (Solo agentes estándar) Amazon MSK le ayuda a supervisar la capacidad de almacenamiento en disco enviándole automáticamente alertas de capacidad de almacenamiento cuando un clúster aprovisionado está a punto de alcanzar su límite de capacidad de almacenamiento. Las alertas también incluyen recomendaciones sobre las mejores medidas que se pueden tomar para solucionar los problemas detectados. Gracias a esta característica, puede identificar y resolver rápidamente los problemas de capacidad del disco antes de que se vuelvan críticos. Amazon MSK envía automáticamente estas alertas a la consola de Amazon MSK, a AWS Health Dashboard Amazon EventBridge y a los contactos de correo electrónico de su AWS cuenta. Para obtener más información sobre las alertas con respecto a la capacidad de almacenamiento, consulte Uso de las alertas de capacidad de almacenamiento de Amazon MSK.

Temas

- Vea las métricas de Amazon MSK mediante CloudWatch
- Métricas de Amazon MSK para monitorear a los corredores estándar con CloudWatch
- Métricas de Amazon MSK para monitorear a los corredores de Express con CloudWatch
- Supervise un clúster aprovisionado de MSK con Prometheus
- Supervisión del desfase del consumidor
- Uso de las alertas de capacidad de almacenamiento de Amazon MSK

Vea las métricas de Amazon MSK mediante CloudWatch

Puede supervisar las métricas de Amazon MSK mediante la CloudWatch consola, la línea de comandos o la CloudWatch API. Los siguientes procedimientos le muestran cómo obtener acceso a las métricas a través de los distintos métodos descritos a continuación.

Para acceder a las métricas mediante la consola CloudWatch

Inicie sesión en AWS Management Console y abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.

1. En el panel de navegación, seleccione Métricas.
2. Seleccione la pestaña Todas las métricas y, a continuación, seleccione AWS/Kafka.
3. Para ver métricas a nivel de tema, elija Topic, Broker ID, Cluster Name (Tema, Identificador de Agente, Nombre de Clúster); para métricas a nivel de agente, elija Broker ID, Cluster name (Identificador de agente, Nombre de clúster); y para métricas a nivel de clúster, elija Cluster Name (Nombre de clúster).
4. (Opcional) En el panel gráfico, seleccione una estadística y un período de tiempo y, a continuación, cree una CloudWatch alarma con estos ajustes.

Para acceder a las métricas mediante el AWS CLI

Utilice las métricas y get-metric-statisticslos comandos de la lista.

Para acceder a las métricas mediante la CloudWatch CLI

Utilice los comandos mon-list-metrics y mon-get-stats.

Para acceder a las métricas mediante la CloudWatch API

Utilice las operaciones ListMetrics y GetMetricStatistics.

Métricas de Amazon MSK para monitorear a los corredores estándar con CloudWatch

Amazon MSK se integra con Amazon CloudWatch para que pueda recopilar, ver y analizar CloudWatch las métricas de sus corredores estándar de MSK. Las métricas que configure para los clústeres aprovisionados por MSK se recopilan automáticamente y se actualizan CloudWatch en intervalos de 1 minuto. Puede establecer el nivel de supervisión de un clúster aprovisionado de MSK en uno de los siguientes:`DEFAULT`,, `PER_BROKER` o. `PER_TOPIC_PER_BROKER` `PER_TOPIC_PER_PARTITION` Las tablas de las siguientes secciones muestran todas las métricas que hay disponibles a partir de cada nivel de supervisión.

> **Note**
>
> Los nombres de algunas métricas de Amazon MSK para la CloudWatch supervisión han cambiado en la versión 3.6.0 y versiones posteriores. Use los nuevos nombres para supervisar estas métricas. En el caso de las métricas con nombres modificados, en la siguiente tabla se muestra el nombre utilizado en la versión 3.6.0 y las versiones posteriores, seguido del nombre en la versión 2.8.2.tiered.

Las métricas de DEFAULT son gratuitas. Los precios de otras métricas se describen en la página de CloudWatchprecios de Amazon.

Supervisión de **DEFAULT**

Las métricas descritas en la tabla siguiente están disponibles en el nivel de monitorización DEFAULT. Son gratis.

Nombre	Cuando está visible	Dimensiones	Descripción
ActiveControllerCount	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster	Sólo debe estar activo en un momento dado un controlador por clúster.
BurstBalance	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El saldo restante de la ráfaga de entrada y salida se destina a los volúmenes de EBS del clúster. Úselo para investigar la latencia o la disminución del rendimiento. BurstBalance no se registra para los volúmenes de EBS cuando el rendimiento de la referencia de un volumen es mayor que el rendimiento por ráfagas máximo. Para obtener más información, consulte Créditos de E/S y rendimiento por ráfagas.
BytesInPerSec	Después de crear un tema.	Nombre del clúster, ID del agente, Tema	El número de bytes por segundo recibidos de los clientes. Esta métrica está disponible por agente y también por tema.
BytesOutPerSec	Después de crear un tema.	Nombre del clúster,	El número de bytes por segundo enviados a los clientes. Esta métrica

Nombre	Cuando está visible	Dimensiones	Descripción
		ID del agente, Tema	está disponible por agente y también por tema.
`ClientConnectionCount`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente, autenticación del cliente	El número de conexiones de cliente autenticadas y activas.
`ConnectionCount`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de conexiones activas autenticadas, no autenticadas y entre agentes.

Nombre	Cuando está visible	Dimensiones	Descripción
`CPUCreditBalance`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	La cantidad de créditos de la CPU obtenidos que un agente ha acumulado desde que se lanzó. Los créditos se acumulan en el saldo de créditos una vez obtenidos y se eliminan del saldo de créditos cuando se gastan. Si se queda sin saldo de créditos de CPU, puede repercutir negativamente en el rendimiento del clúster. Puede tomar medidas para reducir la carga de la CPU. Por ejemplo, puede reducir el número de solicitudes de los clientes o actualizar el tipo de agente a un tipo de agente M5.
`CpuIdle`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El porcentaje de tiempo de inactividad de la CPU.
`CpuIoWait`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El porcentaje de tiempo de inactividad de la CPU durante una operación de disco pendiente.
`CpuSystem`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El porcentaje de CPU en el espacio del kernel.

Nombre	Cuando está visible	Dimensiones	Descripción
`CpuUser`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El porcentaje de CPU en el espacio de usuario.
`GlobalPartitionCount`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster	El número de particiones en todos los temas del clúster, sin incluir las réplicas. Como `GlobalPartitionCount` no incluye réplicas, la suma de los `PartitionCount` valores puede ser mayor que GlobalPartitionCount si el factor de replicación de un tema es superior a 1.
`GlobalTopicCount`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster	Número total de temas entre todos los agentes del clúster.
`EstimatedMaxTimeLag`	Después de que el grupo de consumidores consuma de un tema.	Nombre del clúster, grupo de consumidores, tema	Tiempo estimado (en segundos) para drenar `MaxOffsetLag` .
`KafkaAppLogsDiskUsed`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	Porcentaje de espacio en disco utilizado para los registros de aplicaciones.

Nombre	Cuando está visible	Dimensiones	Descripción
`KafkaDataLogsDiskUsed (Cluster Name, Broker ID` dimensión)	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	Porcentaje de espacio en disco utilizado para los registros de datos.
`LeaderCount`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número total de líderes de particiones por agente, sin incluir las réplicas.
`MaxOffsetLag`	Después de que el grupo de consumidores consuma de un tema.	Nombre del clúster, grupo de consumidores, tema	El retraso máximo de desplazamiento en todas las particiones de un tema.
`MemoryBuffered`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El tamaño en bytes de memoria almacenada en búfer para el agente.
`MemoryCached`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El tamaño en bytes de memoria almacenada en caché para el agente..

Nombre	Cuando está visible	Dimensiones	Descripción
MemoryFree	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El tamaño en bytes de memoria que está libre y disponible para el agente.
HeapMemoryAfterGC	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El porcentaje de memoria apilada total que se utiliza después de la recopilación de elementos no utilizados.
MemoryUsed	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El tamaño en bytes de memoria que está en uso para el agente.
MessagesInPerSec	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de mensajes entrantes por segundo para el agente.
NetworkRxDropped	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de paquetes abandonados descartados.
NetworkRxErrors	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de errores de recepción de la red para el agente.

Nombre	Cuando está visible	Dimensiones	Descripción
`NetworkRxPackets`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de paquetes recibidos por el agente.
`NetworkTxDropped`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de paquetes de transmisión descartados.
`NetworkTxErrors`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de errores de transmisión de red para el agente.
`NetworkTxPackets`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de paquetes transmitidos por el agente.
`OfflinePartitionsCount`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster	Número total de particiones sin conexión en el clúster.
`PartitionCount`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número total de particiones de temas por agente, lo que incluye las réplicas.

Nombre	Cuando está visible	Dimensiones	Descripción
`ProduceTotalTimeMsMean`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El tiempo medio de producción en milisegundos.
`RequestBytesMean`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número medio de bytes de solicitud para el agente.
`RequestTime`	Después de aplicar la limitación de solicitudes.	Nombre del clúster, ID del agente	El tiempo promedio en milisegundos empleado en la red de agentes y subprocesos de E/S para procesar solicitudes.
`RootDiskUsed`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El porcentaje del disco raíz utilizado por el agente.
`SumOffsetLag`	Después de que el grupo de consumidores consuma de un tema.	Nombre del clúster, grupo de consumidores, tema	El retraso de desplazamiento agregado de todas las particiones de un tema.

Nombre	Cuando está visible	Dimensiones	Descripción
SwapFree	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El tamaño en bytes de memoria de intercambio que está disponible para el agente.
SwapUsed	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El tamaño en bytes de memoria de intercambio que está en uso para el agente.
TrafficShaping	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	Métricas de alto nivel que indican la cantidad de paquetes formados (descartados o en cola) debido a que se superan las asignaciones de red. Las métricas PER_BROKER ofrecen información más detallada.
UnderMinIsrPartitionCount	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de particiones bajo minIsr para el agente.
UnderReplicatedPartitions	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de particiones infrarepl icadas para el agente.

Nombre	Cuando está visible	Dimensiones	Descripción
`UserPartitionExists`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	Métrica booleana que indica la presencia de una partición propiedad del usuario en un bróker. Un valor de 1 indica la presencia de particiones en el bróker.
`ZooKeeperRequestLatencyMsMean`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	Para un clúster ZooKeeper basado. La latencia media en milisegundos de ZooKeeper las solicitudes de Apache del bróker.
`ZooKeeperSessionState`	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	Para un clúster ZooKeeper basado. El estado de conexión de la ZooKeeper sesión del corredor puede ser uno de los siguientes: NOT_CONNECTED: '0.0', ASSOCIATING: '0.1', CONNECTING: '0.5', CONNECTEDREADONLY: '0.8', CONNECTED: '1.0', CLOSED: '5.0', AUTH_FAILED: '10.0'.

Supervisión de **`PER_BROKER`**

Al establecer el nivel de supervisión en `PER_BROKER`, obtendrá las métricas descritas en la tabla siguiente además de todas las métricas a nivel de `DEFAULT`. Usted paga por las métricas de la tabla siguiente, mientras que las métricas a nivel `DEFAULT` siguen siendo gratuitas. Las métricas de esta tabla tienen las dimensiones siguientes: nombre del clúster, identificador del agente.

Nombre	Cuando está visible	Descripción
`BwInAllowanceExceeded`	Después de que el clúster llegue	El número de paquetes formados porque el ancho de banda agregado entrante superó el máximo del agente.

Nombre	Cuando está visible	Descripción
	al estado ACTIVE (Activo).	
BwOutAllowanceExceeded	Después de que el clúster llegue al estado ACTIVE (Activo).	El número de paquetes formados porque el ancho de banda agregado saliente superó el máximo del agente.
ConntrackAllowanceExceeded	Después de que el clúster llegue al estado ACTIVE (Activo).	El número de paquetes formados porque el seguimiento de la conexión superó el máximo del agente. El seguimiento de conexiones está relacionado con los grupos de seguridad que hacen un seguimiento de cada conexión establecida para asegurarse de que los paquetes devueltos se entreguen como se espera.
ConnectionCloseRate	Después de que el clúster llegue al estado ACTIVE (Activo).	El número de conexiones cerradas por segundo por oyente. Este número se agrega por oyente y se filtra para los oyentes del cliente.
ConnectionCreationRate	Después de que el clúster llegue al estado ACTIVE (Activo).	El número de conexiones nuevas establecidas por segundo por oyente. Este número se agrega por oyente y se filtra para los oyentes del cliente.

Nombre	Cuando está visible	Descripción
CpuCreditUsage	Después de que el clúster llegue al estado ACTIVE (Activo).	La cantidad de créditos de CPU empleados por el agente. Si se queda sin saldo de créditos de CPU, puede repercutir negativamente en el rendimiento del clúster. Puede tomar medidas para reducir la carga de la CPU. Por ejemplo, puede reducir el número de solicitudes de los clientes o actualizar el tipo de agente a un tipo de agente M5.
FetchConsumerLocalTimeMsMean	Después de que haya un productor/consumidor.	Tiempo medio en milisegundos que la solicitud del consumidor se procesa en el líder.
FetchConsumerRequestQueueTimeMsMean	Después de que haya un productor/consumidor.	Tiempo medio en milisegundos que la solicitud del consumidor espera en la cola de solicitudes.
FetchConsumerResponseQueueTimeMsMean	Después de que haya un productor/consumidor.	Tiempo medio en milisegundos que la solicitud del consumidor espera en la cola de respuesta.
FetchConsumerResponseSendTimeMsMean	Después de que haya un productor/consumidor.	Tiempo medio en milisegundos para que el consumidor envíe una respuesta.
FetchConsumerTotalTimeMsMean	Después de que haya un productor/consumidor.	El tiempo total medio en milisegundos que los consumidores gastan en obtener datos del agente.
FetchFollowerLocalTimeMsMean	Después de que haya un productor/consumidor.	Tiempo medio en milisegundos que la solicitud del seguidor se procesa en el líder.

Nombre	Cuando está visible	Descripción
FetchFollowerRequestQueueTimeMsMean	Después de que haya un productor/consumidor.	Tiempo medio en milisegundos que la solicitud del seguidor espera en la cola de solicitudes.
FetchFollowerResponseQueueTimeMsMean	Después de que haya un productor/consumidor.	Tiempo medio en milisegundos que la solicitud del seguidor espera en la cola de respuesta.
FetchFollowerResponseSendTimeMsMean	Después de que haya un productor/consumidor.	Tiempo medio en milisegundos para que el seguidor envíe una respuesta.
FetchFollowerTotalTimeMsMean	Después de que haya un productor/consumidor.	El tiempo total medio en milisegundos que los seguidores gastan en obtener datos del agente.
FetchMessageConversionsPerSec	Después de crear un tema.	El número de conversiones de mensajes de recuperación por segundo para el agente.
FetchThrottleByteRate	Después de aplicar la limitación del ancho de banda.	El número de bytes acelerados por segundo.
FetchThrottleQueueSize	Después de aplicar la limitación del ancho de banda.	El número de mensajes en la cola del acelerador.
FetchThrottleTime	Después de aplicar la limitación del ancho de banda.	El tiempo promedio de aceleración de recuperación en milisegundos.
IAMNumberOfConnectionRequests	Después de que el clúster llegue al estado ACTIVE (Activo).	La cantidad de solicitudes de autenticación de IAM por segundo.

Nombre	Cuando está visible	Descripción
IAMTooManyConnections	Después de que el clúster llegue al estado ACTIVE (Activo).	La cantidad de conexiones intentadas por encima de 100. 0 indica que la cantidad de conexiones está dentro del límite. Si es menor que 0, se está excediendo el límite de aceleración y es necesario reducir la cantidad de conexiones.
NetworkProcessorAvgIdlePercent	Después de que el clúster llegue al estado ACTIVE (Activo).	Porcentaje medio del tiempo en que los procesadores de red están inactivos.
PpsAllowanceExceeded	Después de que el clúster llegue al estado ACTIVE (Activo).	El número de paquetes formados porque el PPS bidireccional superó el máximo del agente.
ProduceLocalTimeMsMean	Después de que el clúster llegue al estado ACTIVE (Activo).	Tiempo medio en milisegundos en el que la solicitud se procesa en el líder.
ProduceMessageConversionsPerSec	Después de crear un tema.	El número de conversiones de mensajes de generación por segundo para el agente.
ProduceMessageConversionsTimeMsMean	Después de que el clúster llegue al estado ACTIVE (Activo).	Tiempo medio en milisegundos invertido en conversiones de formato de mensaje.
ProduceRequestQueueTimeMsMean	Después de que el clúster llegue al estado ACTIVE (Activo).	Tiempo medio en milisegundos que los mensajes de solicitud pasan en la cola.

Nombre	Cuando está visible	Descripción
`ProduceResponseQueueTimeMsMean`	Después de que el clúster llegue al estado ACTIVE (Activo).	Tiempo medio en milisegundos que pasan los mensajes de respuesta en la cola.
`ProduceResponseSendTimeMsMean`	Después de que el clúster llegue al estado ACTIVE (Activo).	Tiempo medio en milisegundos dedicado al envío de mensajes de respuesta.
`ProduceThrottleByteRate`	Después de aplicar la limitación del ancho de banda.	El número de bytes acelerados por segundo.
`ProduceThrottleQueueSize`	Después de aplicar la limitación del ancho de banda.	El número de mensajes en la cola del acelerador.
`ProduceThrottleTime`	Después de aplicar la limitación del ancho de banda.	El tiempo promedio de producción del acelerador en milisegundos.
`ProduceTotalTimeMsMean`	Después de que el clúster llegue al estado ACTIVE (Activo).	El tiempo medio de producción en milisegundos.
`RemoteFetchBytesPerSec (RemoteBytesInPerSec in v2.8.2.tiered)`	Después de que haya un productor/consumidor.	El número total de bytes transferidos desde el almacenamiento por niveles en respuesta a las búsquedas de los consumidores. Esta métrica incluye todas las particiones de temas que contribuyen al tráfico de transferencia de datos descendente. Categoría: tasas de tráfico y error. Se trata de una métrica KIP-405.

Nombre	Cuando está visible	Descripción
`RemoteCopyBytesPerSec (RemoteBytesOutPerSec in v2.8.2.tiered)`	Después de que haya un productor/consumidor.	El número total de bytes transferidos al almacenamiento por niveles, incluidos los datos de los segmentos de registro, los índices y otros archivos auxiliares. Esta métrica incluye todas las particiones de temas que contribuyen al tráfico ascendente de transferencia de datos. Categoría: tasas de tráfico y error. Se trata de una métrica KIP-405.
`RemoteLogManagerTasksAvgIdlePercent`	Después de que el clúster llegue al estado ACTIVE (Activo).	El porcentaje medio de tiempo que el administrador de registros remoto pasó inactivo. El administrador de registros remoto transfiere los datos del agente al almacenamiento por niveles. Categoría: actividad interna. Se trata de una métrica KIP-405.
`RemoteLogReaderAvgIdlePercent`	Después de que el clúster llegue al estado ACTIVE (Activo).	El porcentaje medio de tiempo que el lector de registros remoto pasó inactivo. El lector de registros remoto transfiere los datos del almacenamiento remoto al agente en respuesta a las solicitudes de los consumidores. Categoría: actividad interna. Se trata de una métrica KIP-405.
`RemoteLogReaderTaskQueueSize`	Después de que el clúster llegue al estado ACTIVE (Activo).	El número de tareas responsables de las lecturas del almacenamiento por niveles que están pendientes de ser programadas. Categoría: actividad interna. Se trata de una métrica KIP-405.

Nombre	Cuando está visible	Descripción
`RemoteFetchErrorsPerSec (RemoteReadErrorPerSec in v2.8.2.tiered)`	Después de que el clúster llegue al estado ACTIVE (Activo).	La tasa total de errores en respuesta a las solicitudes de lectura que el agente especificado envió al almacenamiento por niveles para recuperar datos en respuesta a las búsquedas de los consumidores. Esta métrica incluye todas las particiones de temas que contribuyen al tráfico de transferencia de datos descendente. Categoría: tasas de tráfico y error. Se trata de una métrica KIP-405.
`RemoteFetchRequestsPerSec (RemoteReadRequestsPerSec in v2.8.2.tiered)`	Después de que el clúster llegue al estado ACTIVE (Activo).	El número total de solicitudes de lectura que el agente especificado envió al almacenamiento por niveles para recuperar datos en respuesta a las búsquedas de los consumidores. Esta métrica incluye todas las particiones de temas que contribuyen al tráfico de transferencia de datos descendente. Categoría: tasas de tráfico y error. Se trata de una métrica KIP-405.
`RemoteCopyErrorsPerSec (RemoteWriteErrorPerSec in v2.8.2.tiered)`	Después de que el clúster llegue al estado ACTIVE (Activo).	La tasa total de errores en respuesta a las solicitudes de escritura que el agente especificado envió al almacenamiento por niveles para transferir datos de forma ascendente. Esta métrica incluye todas las particiones de temas que contribuyen al tráfico ascendente de transferencia de datos. Categoría: tasas de tráfico y error. Se trata de una métrica KIP-405.

Nombre	Cuando está visible	Descripción
`RemoteLogSizeBytes`	Después de que el clúster llegue al estado ACTIVE (Activo).	El número de bytes almacenados en el nivel remoto. Esta métrica está disponible para los clústeres de almacenamiento en niveles de Apache Kafka versión 3.7.x en Amazon MSK.
`ReplicationBytesInPerSec`	Después de crear un tema.	El número de bytes por segundo recibidos de otros agentes.
`ReplicationBytesOutPerSec`	Después de crear un tema.	El número de bytes enviados por segundo a otros agentes.
`RequestExemptFromThrottleTime`	Después de aplicar la limitación de solicitudes.	El tiempo promedio en milisegundos empleado en la red del agente y subprocesos de E/S para procesar solicitudes que están exentas de limitación.
`RequestHandlerAvgIdlePercent`	Después de que el clúster llegue al estado ACTIVE (Activo).	El porcentaje medio del tiempo en que los subprocesos del controlador de solicitudes están inactivos.
`RequestThrottleQueueSize`	Después de aplicar la limitación de solicitudes.	El número de mensajes en la cola del acelerador.
`RequestThrottleTime`	Después de aplicar la limitación de solicitudes.	El tiempo medio de aceleración de la solicitud en milisegundos.
`TcpConnections`	Después de que el clúster llegue al estado ACTIVE (Activo).	Muestra el número de segmentos TCP entrantes y salientes con el indicador SYN establecido.

Nombre	Cuando está visible	Descripción
`RemoteCopyLagBytes (TotalTierBytesLag in v2.8.2.tiered)`	Después de crear un tema.	El número total de bytes de los datos que son aptos para la organización en niveles en el agente, pero que aún no se han transferido al almacenamiento por niveles. Estas métricas muestran la eficiencia de la transferencia de datos ascendentes. A medida que aumenta el desfase, aumenta la cantidad de datos que no permanecen en el almacenamiento por niveles. Categoría: desfase del archivo. No se trata de una métrica KIP-405.
`TrafficBytes`	Después de que el clúster llegue al estado ACTIVE (Activo).	Muestra el tráfico de red en bytes totales entre clientes (productores y consumidores) y agentes. No se informa del tráfico entre agentes.
`VolumeQueueLength`	Después de que el clúster llegue al estado ACTIVE (Activo).	El número de solicitudes de operaciones de lectura y escritura a la espera de realizarse en un periodo de tiempo especificado.
`VolumeReadBytes`	Después de que el clúster llegue al estado ACTIVE (Activo).	El número de bytes indexados leídos en un periodo de tiempo especificado.
`VolumeReadOps`	Después de que el clúster llegue al estado ACTIVE (Activo).	El número total de operaciones de lectura realizadas en un periodo de tiempo especificado.

Nombre	Cuando está visible	Descripción
`VolumeTotalReadTime`	Después de que el clúster llegue al estado ACTIVE (Activo).	Número total de segundos empleados por todas las operaciones de lectura que se realizaron en un periodo de tiempo especificado.
`VolumeTotalWriteTime`	Después de que el clúster llegue al estado ACTIVE (Activo).	Número total de segundos empleados por todas las operaciones de escritura que se realizaron en un periodo de tiempo especificado.
`VolumeWriteBytes`	Después de que el clúster llegue al estado ACTIVE (Activo).	El número de bytes escritos en un periodo de tiempo especificado.
`VolumeWriteOps`	Después de que el clúster llegue al estado ACTIVE (Activo).	El número total de operaciones de escritura en un periodo especificado.

Supervisión de **`PER_TOPIC_PER_BROKER`**

Al establecer el nivel de supervisión en `PER_TOPIC_PER_BROKER`, obtendrá las métricas descritas en la tabla siguiente, además de todas las métricas de los niveles `PER_BROKER` y DEFAULT (Predeterminado). Solo las métricas de nivel `DEFAULT` son gratuitas. Las métricas de esta tabla tienen las dimensiones siguientes: nombre del clúster, identificador del agente, tema.

> **Important**
>
> En el caso de un clúster de Amazon MSK que utiliza la versión 2.4.1 o posterior de Apache Kafka, las métricas de la siguiente tabla aparecen solo después de que sus valores sean distintos de cero por primera vez. Por ejemplo, para ver `BytesInPerSec`, uno o más productores deben enviar datos al clúster en primer lugar.

Nombre	Cuando está visible	Descripción
`FetchMessageConversionsPerSec`	Después de crear un tema.	El número de mensajes recuperados convertidos por segundo.
`MessagesInPerSec`	Después de crear un tema.	El número de mensajes recibidos por segundo.
`ProduceMessageConversionsPerSec`	Después de crear un tema.	El número de conversiones por segundo para los mensajes producidos.
`RemoteFetchBytesPerSec (RemoteBytesInPerSec in v2.8.2.tiered)`	Después de crear un tema y de que el tema esté produciendo o consumiendo.	El número de bytes transferidos desde el almacenamiento por niveles en respuesta a las búsquedas del consumidor por el tema y el agente especificados. Esta métrica incluye todas las particiones de temas que contribuyen al tráfico de transferencia de datos descendente en el agente especificado. Categoría: tasas de tráfico y error. Se trata de una métrica KIP-405.
`RemoteCopyBytesPerSec (RemoteBytesOutPerSec in v2.8.2.tiered)`	Después de crear un tema y de que el tema esté produciendo o consumiendo.	El número de bytes transferidos al almacenamiento por niveles, para el tema y el agente especificados. Esta métrica incluye todas las particiones del tema que contribuyen al tráfico ascendente de transferencia de datos en el agente especificado. Categoría: tasas de tráfico y error. Se trata de una métrica KIP-405.
`RemoteFetchErrorsPerSec (RemoteReadErrorPerSec in v2.8.2.tiered)`	Después de crear un tema y de que el tema esté produciendo o consumiendo.	La tasa de errores en respuesta a las solicitudes de lectura que el agente especificado envía al almacenamiento por niveles para recuperar datos en respuesta a las consultas de los consumidores sobre el tema especificado. Esta métrica incluye todas las particiones de temas que contribuyen al tráfico de transferencia de datos descendente en el agente especificado.

Nombre	Cuando está visible	Descripción
		Categoría: tasas de tráfico y error. Se trata de una métrica KIP-405.
`RemoteFetchRequestsPerSec (RemoteReadRequestsPerSec in v2.8.2.tiered)`	Después de crear un tema y de que el tema esté produciendo o consumiendo.	El número de solicitudes de lectura que el agente específico envía al almacenamiento por niveles para recuperar datos en respuesta a las consultas de los consumidores sobre el tema especificado. Esta métrica incluye todas las particiones de temas que contribuyen al tráfico de transferencia de datos descendente en el agente especificado. Categoría: tasas de tráfico y error. Se trata de una métrica KIP-405.
`RemoteCopyErrorsPerSec (RemoteWriteErrorPerSec in v2.8.2.tiered)`	Después de crear un tema y de que el tema esté produciendo o consumiendo.	La tasa de errores en respuesta a las solicitudes de escritura que el agente especificado envía al almacenamiento por niveles para transferir datos de forma ascendente. Esta métrica incluye todas las particiones del tema que contribuyen al tráfico ascendente de transferencia de datos en el agente especificado. Categoría: tasas de tráfico y error. Se trata de una métrica KIP-405.
`RemoteLogSizeBytes`	Después de crear un tema.	El número de bytes almacenados en el nivel remoto. Esta métrica está disponible para los clústeres de almacenamiento en niveles de Apache Kafka versión 3.7.x en Amazon MSK.

Supervisión de **`PER_TOPIC_PER_PARTITION`**

Al establecer el nivel de supervisión en `PER_TOPIC_PER_PARTITION`, obtendrá las métricas descritas en la tabla siguiente, además de todas las métricas de los niveles `PER_TOPIC_PER_BROKER`, `PER_BROKER` y DEFAULT. Solo las métricas de nivel `DEFAULT` son

gratuitas. Las métricas de esta tabla tienen las siguientes dimensiones: grupo de consumidores, tema y partición.

Nombre	Cuando está visible	Descripción
`EstimatedTimeLag`	Después de que el grupo de consumidores consuma de un tema.	Tiempo estimado (en segundos) para reducir el retraso de desplazamiento de la partición.
`OffsetLag`	Después de que el grupo de consumidores consuma de un tema.	El desfase del consumidor de la partición en cuanto al número de compensaciones.

Conozca los estados de los clústeres aprovisionados por MSK

En la siguiente tabla se muestran los posibles estados de un clúster aprovisionado de MSK y se describe su significado. A menos que se especifique lo contrario, los estados de los clústeres aprovisionados de MSK se aplican a los tipos de bróker Standard y Express. En esta tabla también se describen las acciones que puede y no puede realizar cuando un clúster aprovisionado de MSK se encuentra en uno de estos estados. Para conocer el estado de un clúster, puede visitar la AWS Management Console. También puede usar el comando describe-cluster-v2 o la operación DescribeClusterV2 para describir el clúster aprovisionado. La descripción de un clúster incluye su estado.

Estado del clúster aprovisionado de MSK	Significado y posibles acciones
ACTIVE	Puede producir y consumir datos. También puede realizar AWS CLI operaciones y API de Amazon MSK en el clúster.
CREAR	Amazon MSK está configurando el clúster aprovisionado. Debe esperar a que el clúster

Estado del clúster aprovisionado de MSK	Significado y posibles acciones
	alcance el estado ACTIVO antes de poder usarlo para producir o consumir datos o para realizar AWS CLI operaciones o la API de Amazon MSK en él.
ELIMINANDO	Se está eliminando el clúster aprovisionado. No puede usarlo para producir o consumir datos. Tampoco puede realizar AWS CLI operaciones ni la API de Amazon MSK en ella.
ERROR	Se produjo un error en el proceso de creación o eliminación del clúster aprovisionado. No puede usar el clúster para producir o consumir datos. Puede eliminar el clúster, pero no puede realizar operaciones de AWS CLI actualización ni API de Amazon MSK en él.
HEALING	Amazon MSK está llevando a cabo una operación interna, como reemplazar a un agente en mal estado. Por ejemplo, es posible que el agente no responda. Puede seguir utilizando el clúster aprovisionado para producir y consumir datos. Sin embargo, no puede realizar operaciones de API ni de AWS CLI actualización de Amazon MSK en el clúster hasta que vuelva al estado ACTIVO.

Estado del clúster aprovisionado de MSK	Significado y posibles acciones
MAINTENANCE	(Solo agentes estándar) Amazon MSK realiza operaciones de mantenimiento rutinarias en el clúster. Estas operaciones de mantenimi ento incluyen la aplicación de parches de seguridad. Aún puede usar el clúster para producir y consumir datos. Sin embargo, no puede realizar operaciones de actualización de la API o AWS CLI de Amazon MSK en el clúster hasta que vuelva al estado ACTIVO. El estado del clúster permanece ACTIVO durante el mantenimiento en los corredores Express. Consulte Aplicación de parches.
REBOOTING_BROKER	Amazon MSK está reiniciando un agente. Puede seguir utilizando el clúster aprovisionado para producir y consumir datos. Sin embargo, no puede realizar operaciones de API ni de AWS CLI actualización de Amazon MSK en el clúster hasta que vuelva al estado ACTIVO.
ACTUALIZANDO	Una AWS CLI operación o API de Amazon MSK iniciada por el usuario está actualiza ndo el clúster aprovisionado. Puede seguir utilizando el clúster aprovisionado para producir y consumir datos. Sin embargo, no puede realizar ninguna operación adicional de API o AWS CLI actualización de Amazon MSK en el clúster hasta que vuelva al estado ACTIVO.

Métricas de Amazon MSK para monitorear a los corredores de Express con CloudWatch

Amazon MSK se integra CloudWatch para que pueda recopilar, ver y analizar CloudWatch las métricas de sus corredores de MSK Express. Las métricas que configure para los clústeres

aprovisionados de MSK se recopilan automáticamente y se actualizan CloudWatch en intervalos de 1 minuto. Puede establecer el nivel de supervisión de un clúster aprovisionado de MSK en uno de los siguientes:`DEFAULT`,, `PER_BROKER` o. `PER_TOPIC_PER_BROKER` `PER_TOPIC_PER_PARTITION` En las tablas de las siguientes secciones se muestran las métricas disponibles a partir de cada nivel de supervisión.

Las métricas de `DEFAULT` son gratuitas. Los precios de otras métricas se describen en la página de CloudWatchprecios de Amazon.

`DEFAULT`Monitorización de niveles para corredores de Express

Las métricas descritas en la tabla siguiente están disponibles en el nivel de monitorización `DEFAULT`. Son gratis.

Monitorización de nivel PREDETERMINADO para corredores de Express

Nombre	Cuando está visible	Dimensiones	Descripción
ActiveControllerCount	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster	Sólo debe estar activo en un momento dado un controlador por clúster.
BytesInPerSec	Después de crear un tema.	Nombre del clúster, ID del agente, Tema	El número de bytes por segundo recibidos de los clientes. Esta métrica está disponible por agente y también por tema.
BytesOutPerSec	Después de crear un tema.	Nombre del clúster, ID del agente, Tema	El número de bytes por segundo enviados a los clientes. Esta métrica está disponibl e por agente y también por tema.
ClientConnectionCo unt	Después de que el clúster llegue	Nombre del clúster, ID del agente, autentica ción del cliente	El número de conexiones de

Nombre	Cuando está visible	Dimensiones	Descripción
	al estado ACTIVE (Activo).		cliente autenticadas y activas.
ConnectionCount	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de conexiones activas autenticadas, no autenticadas y entre agentes.
CpuIdle	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El porcentaje de tiempo de inactividad de la CPU.
CpuSystem	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El porcentaje de CPU en el espacio del kernel.
CpuUser	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El porcentaje de CPU en el espacio de usuario.

Nombre	Cuando está visible	Dimensiones	Descripción
GlobalPartitionCount	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster	El número de particiones en todos los temas del clúster, sin incluir las réplicas. Como `GlobalPartitionCount` no incluye réplicas, la suma de los `PartitionCount` valores puede ser mayor que `GlobalPartitionCount` si el factor de replicación de un tema es superior 1 a.
GlobalTopicCount	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster	Número total de temas entre todos los agentes del clúster.
EstimatedMaxTimeLag	Después de que el grupo de consumidores consuma de un tema.	Grupo de consumidores, tema	Tiempo estimado (en segundos) para drenar `MaxOffsetLag` .
LeaderCount	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número total de líderes de particiones por agente, sin incluir las réplicas.

Nombre	Cuando está visible	Dimensiones	Descripción
MaxOffsetLag	Después de que el grupo de consumidores consuma de un tema.	Grupo de consumidores, tema	El retraso máximo de desplazamiento en todas las particiones de un tema.
MemoryBuffered	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El tamaño en bytes de memoria almacenada en búfer para el agente.
MemoryCached	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El tamaño en bytes de memoria almacenada en caché para el agente..
MemoryFree	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El tamaño en bytes de memoria que está libre y disponible para el agente.
MemoryUsed	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El tamaño en bytes de memoria que está en uso para el agente.
MessagesInPerSec	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de mensajes entrantes por segundo para el agente.
NetworkRxDropped	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de paquetes abandonados descartados.

Nombre	Cuando está visible	Dimensiones	Descripción
NetworkRxErrors	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de errores de recepción de la red para el agente.
NetworkRxPackets	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de paquetes recibidos por el agente.
NetworkTxDropped	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de paquetes de transmisión descartados.
NetworkTxErrors	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de errores de transmisión de red para el agente.
NetworkTxPackets	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de paquetes transmitidos por el agente.
PartitionCount	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número total de particiones de temas por agente, lo que incluye las réplicas.
ProduceTotalTimeMs Mean	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El tiempo medio de producción en milisegundos.

Nombre	Cuando está visible	Dimensiones	Descripción
RequestBytesMean	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número medio de bytes de solicitud para el agente.
RequestTime	Después de aplicar la limitación de solicitudes.	Nombre del clúster, ID del agente	El tiempo medio en milisegundos que se pasa en la red de intermediarios y en I/O los subprocesos para procesar las solicitudes.
SumOffsetLag	Después de que el grupo de consumidores consuma de un tema.	Grupo de consumidores, tema	El retraso de desplazamiento agregado de todas las particiones de un tema.
UserPartitionExists	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	Métrica booleana que indica la presencia de una partición propiedad del usuario en un bróker. Un valor de 1 indica la presencia de particiones en el bróker.

`PER_BROKER`Monitorización de niveles para corredores de Express

Al establecer el nivel de supervisión en `PER_BROKER`, obtendrá las métricas descritas en la tabla siguiente además de todas las métricas a nivel de `DEFAULT`. Usted paga por las métricas de la siguiente tabla, mientras que las métricas de `DEFAULT` nivel siguen siendo gratuitas. Las métricas de esta tabla tienen las dimensiones siguientes: nombre del clúster, identificador del agente.

Las métricas adicionales están disponibles a partir del nivel de supervisión de PER_BROKER

Nombre	Cuando está visible	Descripción
ConnectionCloseRate	Después de que el clúster llegue al estado ACTIVE (Activo).	El número de conexiones cerradas por segundo por oyente. Este número se agrega por oyente y se filtra para los oyentes del cliente.
ConnectionCreationRate	Después de que el clúster llegue al estado ACTIVE (Activo).	El número de conexione s nuevas establecidas por segundo por oyente. Este número se agrega por oyente y se filtra para los oyentes del cliente.
FetchConsumerLocal TimeMsMean	Después de que haya un productor/consumidor.	Tiempo medio en milisegun dos que la solicitud del consumidor se procesa en el líder.
FetchConsumerReque stQueueTimeMsMean	Después de que haya un productor/consumidor.	Tiempo medio en milisegun dos que la solicitud del consumidor espera en la cola de solicitudes.
FetchConsumerRespo nseQueueTimeMsMean	Después de que haya un productor/consumidor.	Tiempo medio en milisegun dos que la solicitud del consumidor espera en la cola de respuesta.
FetchConsumerRespo nseSendTimeMsMean	Después de que haya un productor/consumidor.	Tiempo medio en milisegun dos para que el consumidor envíe una respuesta.
FetchConsumerTotal TimeMsMean	Después de que haya un productor/consumidor.	El tiempo total medio en milisegundos que los

Nombre	Cuando está visible	Descripción
		consumidores gastan en obtener datos del agente.
FetchFollowerLocal TimeMsMean	Después de que haya un productor/consumidor.	Tiempo medio en milisegun dos que la solicitud del seguidor se procesa en el líder.
FetchFollowerReque stQueueTimeMsMean	Después de que haya un productor/consumidor.	Tiempo medio en milisegun dos que la solicitud del seguidor espera en la cola de solicitudes.
FetchFollowerRespo nseQueueTimeMsMean	Después de que haya un productor/consumidor.	Tiempo medio en milisegun dos que la solicitud del seguidor espera en la cola de respuesta.
FetchFollowerRespo nseSendTimeMsMean	Después de que haya un productor/consumidor.	Tiempo medio en milisegun dos para que el seguidor envíe una respuesta.
FetchFollowerTotal TimeMsMean	Después de que haya un productor/consumidor.	El tiempo total medio en milisegundos que los seguidores gastan en obtener datos del agente.
FetchThrottleByteRate	Después de aplicar la limitació n del ancho de banda.	El número de bytes acelerado s por segundo.
FetchThrottleQueueSize	Después de aplicar la limitació n del ancho de banda.	El número de mensajes en la cola del acelerador.
FetchThrottleTime	Después de aplicar la limitació n del ancho de banda.	El tiempo promedio de aceleración de recuperación en milisegundos.

Nombre	Cuando está visible	Descripción
IAMNumberOfConnectionRequests	Después de que el clúster llegue al estado ACTIVE (Activo).	La cantidad de solicitudes de autenticación de IAM por segundo.
IAMTooManyConnections	Después de que el clúster llegue al estado ACTIVE (Activo).	El número de conexiones intentadas es superior a 100. `0`significa que el número de conexiones está dentro del límite. Si`>0`, se supera el límite del acelerador y necesita reducir el número de conexiones.
NetworkProcessorAvgIdlePercent	Después de que el clúster llegue al estado ACTIVE (Activo).	Porcentaje medio del tiempo en que los procesadores de red están inactivos.
ProduceLocalTimeMsMean	Después de que el clúster llegue al estado ACTIVE (Activo).	Tiempo medio en milisegundos en el que la solicitud se procesa en el líder.
ProduceRequestQueueTimeMsMean	Después de que el clúster llegue al estado ACTIVE (Activo).	Tiempo medio en milisegundos que los mensajes de solicitud pasan en la cola.
ProduceResponseQueueTimeMsMean	Después de que el clúster llegue al estado ACTIVE (Activo).	Tiempo medio en milisegundos que pasan los mensajes de respuesta en la cola.
ProduceResponseSendTimeMsMean	Después de que el clúster llegue al estado ACTIVE (Activo).	Tiempo medio en milisegundos dedicado al envío de mensajes de respuesta.
ProduceThrottleByteRate	Después de aplicar la limitación del ancho de banda.	El número de bytes acelerados por segundo.

Nombre	Cuando está visible	Descripción
ProduceThrottleQueueSize	Después de aplicar la limitación del ancho de banda.	El número de mensajes en la cola del acelerador.
ProduceThrottleTime	Después de aplicar la limitación del ancho de banda.	El tiempo promedio de producción del acelerador en milisegundos.
ProduceTotalTimeMsMean	Después de que el clúster llegue al estado ACTIVE (Activo).	El tiempo medio de producción en milisegundos.
ReplicationBytesInPerSec	Después de crear un tema.	El número de bytes por segundo recibidos de otros agentes.
ReplicationBytesOutPerSec	Después de crear un tema.	El número de bytes enviados por segundo a otros agentes.
RequestExemptFromThrottleTime	Después de aplicar la limitación de solicitudes.	El tiempo medio en milisegundos que se dedica a la red de intermediarios y a I/O los subprocesos a procesar las solicitudes exentas de limitaciones.
RequestHandlerAvgIdlePercent	Después de que el clúster llegue al estado ACTIVE (Activo).	El porcentaje medio del tiempo en que los subprocesos del controlador de solicitudes están inactivos.
RequestThrottleQueueSize	Después de aplicar la limitación de solicitudes.	El número de mensajes en la cola del acelerador.
RequestThrottleTime	Después de aplicar la limitación de solicitudes.	El tiempo medio de aceleración de la solicitud en milisegundos.

Nombre	Cuando está visible	Descripción
TcpConnections	Después de que el clúster llegue al estado ACTIVE (Activo).	Muestra el número de segmentos TCP entrantes y salientes con el indicador SYN establecido.
TrafficBytes	Después de que el clúster llegue al estado ACTIVE (Activo).	Muestra el tráfico de red en bytes totales entre clientes (productores y consumidores) y agentes. No se informa del tráfico entre agentes.

`PER_TOPIC_PER_PARTITION`monitoreo de nivel para los corredores de Express

Al establecer el nivel de supervisión en`PER_TOPIC_PER_PARTITION`, se obtienen las métricas que se describen en la siguiente tabla, además de todas las métricas de los `DEFAULT` niveles `PER_TOPIC_PER_BROKER`PER_BROKER, y. Solo las métricas de `DEFAULT` nivel son gratuitas. Las métricas de esta tabla tienen las siguientes dimensiones: grupo de consumidores, tema y partición.

Métricas adicionales disponibles a partir del nivel de supervisión PER_PARTITION

Nombre	Cuando está visible	Descripción
EstimatedTimeLag	Después de que el grupo de consumidores consuma de un tema.	Tiempo estimado (en segundos) para reducir el retraso de desplazamiento de la partición.
OffsetLag	Después de que el grupo de consumidores consuma de un tema.	El desfase del consumidor de la partición en cuanto al número de compensaciones.

`PER_TOPIC_PER_BROKER`monitoreo de nivel para corredores de Express

Al establecer el nivel de supervisión en`PER_TOPIC_PER_BROKER`, se obtienen las métricas que se describen en la siguiente tabla, además de todas las métricas de los `DEFAULT` niveles

PER_BROKER y. Solo las métricas de DEFAULT nivel son gratuitas. Las métricas de esta tabla tienen las dimensiones siguientes: nombre del clúster, identificador del agente, tema.

> **Important**
>
> Las métricas de la siguiente tabla aparecen solo después de que sus valores pasen a ser distintos de cero por primera vez. Por ejemplo, para ver BytesInPerSec, uno o más productores deben enviar primero los datos al clúster.

Las métricas adicionales están disponibles a partir del nivel de supervisión PER_TOPIC_PER_BROKER

Nombre	Cuando está visible	Descripción
MessagesInPerSec	Después de crear un tema.	El número de mensajes recibidos por segundo.

Supervise un clúster aprovisionado de MSK con Prometheus

Puede monitorear su clúster aprovisionado de MSK con Prometheus, un sistema de monitoreo de código abierto para datos métricos de series temporales. Puede publicar estos datos en Amazon Managed Service para Prometheus mediante la característica de escritura remota de Prometheus. También puede usar herramientas que sean compatibles con métricas con formato Prometheus o herramientas que se integren con Amazon MSK Open Monitoring, como Datadog, Lenses, New Relic y Sumo logic. El monitoreo abierto está disponible de forma gratuita, pero se aplican cargos por la transferencia de datos a través de las zonas de disponibilidad.

Para obtener información sobre Prometheus, consulte la documentación de Prometheus.

Para obtener información sobre el uso de Prometheus, consulte Mejorar los conocimientos operativos de Amazon MSK mediante Amazon Managed Service for Prometheus y Amazon Managed Grafana.

> **Note**
>
> KRaft el modo de metadatos y los corredores de MSK Express no pueden habilitar la supervisión abierta y el acceso público.

Habilite la supervisión abierta en los nuevos clústeres aprovisionados por MSK

Este procedimiento describe cómo habilitar la supervisión abierta en un nuevo clúster de MSK mediante la AWS Management Console, la o la AWS CLI API de Amazon MSK.

Uso de la AWS Management Console

1. ¿Iniciar sesión en la AWS Management Console consola Amazon MSK y abrirla desde https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. En la sección Monitoring (Monitoreo) marque la casilla de verificación situada junto a Enable open monitoring with Prometheus (Habilitar monitoreo abierta con Prometheus).
3. Proporcione la información requerida en todas las secciones de la página y revise todas las opciones disponibles.
4. Elija Create cluster.

Usando el AWS CLI

- Invoque el comando create-cluster y especifique su opción `open-monitoring`. Habilite `JmxExporter`, `NodeExporter`, o ambos. Si especifica `open-monitoring`, no se pueden desactivar los dos exportadores al mismo tiempo.

Uso de la API

- Invoque la CreateClusteroperación y especifique`OpenMonitoring`. Habilite `jmxExporter`, `nodeExporter`, o ambos. Si especifica `OpenMonitoring`, no se pueden desactivar los dos exportadores al mismo tiempo.

Habilite la supervisión abierta en el clúster aprovisionado de MSK existente

Para habilitar la supervisión abierta, asegúrese de que el clúster aprovisionado de MSK esté en ese estado. `ACTIVE`

Usando el AWS Management Console

1. ¿Iniciar sesión en la AWS Management Console consola Amazon MSK y abrirla desde https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. Elija el nombre del clúster que desea actualizar. Esto le lleva a la página que contiene los detalles del clúster.

3. En la pestaña Propiedades desplácese hacia abajo para buscar la sección Monitoreo.
4. Seleccione Editar.
5. Marque la casilla de verificación situada junto a Enable open monitoring with Prometheus (Habilitar el monitoreo abierto con Prometheus).
6. Seleccione Save changes (Guardar cambios).

Usando el AWS CLI

- Invoque el comando update-monitoring y especifique su opción `open-monitoring`. Habilite `JmxExporter`, `NodeExporter`, o ambos. Si especifica `open-monitoring`, no se pueden desactivar los dos exportadores al mismo tiempo.

Uso de la API

- Invoque la UpdateMonitoringoperación y especifique`OpenMonitoring`. Habilite `jmxExporter`, `nodeExporter`, o ambos. Si especifica `OpenMonitoring`, no se pueden desactivar los dos exportadores al mismo tiempo.

Configurar un host de Prometheus en una instancia de Amazon EC2

En este procedimiento se describe cómo configurar un host de Prometheus mediante un archivo prometheus.yml.

1. Descarga el servidor Prometheus desde tu instancia de https://prometheus.io/download/#prometheus Amazon. EC2
2. Extraiga el archivo descargado en un directorio y vaya a ese directorio.
3. Cree un archivo con el siguiente contenido y llámelo `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. Utilice la ListNodesoperación para obtener una lista de los agentes de su clúster.
5. Cree un archivo llamado `targets.json` con el siguiente JSON. Sustituya *broker_dns_1broker_dns_2*, y el resto de los nombres DNS del corredor por los nombres DNS que obtuvo para sus corredores en el paso anterior. Incluya todos los agentes que obtuvo en el paso anterior. Amazon MSK utiliza el puerto 11001 para el exportador de JMX y el puerto 11002 para el exportador de nodos.

   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**
>
> Para extraer las métricas JMX de KRaft los controladores, añade los nombres DNS de los controladores como objetivos en el archivo JSON. Por ejemplo: `controller_dns_1:11001`, en donde debe reemplazar `controller_dns_1` con el nombre de DNS real del controlador.

6. Para iniciar el servidor Prometheus en su instancia de EC2 Amazon, ejecute el siguiente comando en el directorio en el que extrajo los archivos de Prometheus y guardó y. `prometheus.yml` `targets.json`

```
./prometheus
```

7. Busca la dirección IP IPv4 pública de la EC2 instancia de Amazon en la que ejecutaste Prometheus en el paso anterior. Necesitará esta dirección IP pública en el siguiente paso.
8. Para acceder a la interfaz de usuario web de Prometheus, abre un navegador que pueda acceder a tu instancia de EC2 Amazon y ve *`Prometheus-Instance-Public-IP`*`:9090` a, *`Prometheus-Instance-Public-IP`* donde está la dirección IP pública que obtuviste en el paso anterior.

Uso de las métricas de Prometheus

Se puede acceder a todas las métricas emitidas por Apache Kafka a JMX mediante el monitoreo abierto con Prometheus. Para obtener información acerca de las métricas de Apache Kafka, consulte Monitoring en la documentación de Apache Kafka. Junto con las métricas de Apache Kafka, las métricas de retardo de consumo también están disponibles en el puerto 11001 con el nombre JMX. MBean `kafka.consumer.group:type=ConsumerLagMetrics` También puede usar el exportador de nodos de Prometheus para obtener métricas de CPU y disco para sus agentes en el puerto 11002.

Almacenamiento de las métricas de Prometheus en Amazon Managed Service para Prometheus

Amazon Managed Service para Prometheus es un servicio de supervisión y alertas compatible con Prometheus que puede utilizar para supervisar clústeres de Amazon MSK. Es un servicio totalmente administrado que escala automáticamente la ingesta, el almacenamiento, las consultas y las alertas de sus métricas. También se integra con los servicios AWS de seguridad para ofrecerle un acceso

rápido y seguro a sus datos. Puede utilizar el lenguaje de consulta ProMQL de código abierto para consultar sus métricas y crear alertas sobre ellas.

Para obtener más información, consulte Primeros pasos con Amazon Managed Service for Prometheus.

Supervisión del desfase del consumidor

Supervisar el desfase del consumidor permite identificar a los consumidores lentos o estancados que no están al tanto de los últimos datos disponibles sobre un tema. Cuando sea necesario, puede tomar medidas correctivas, como ampliar el número de consumidores o reiniciarlos. Para monitorear el retraso de consumo, puedes usar Amazon CloudWatch o el monitoreo abierto con Prometheus.

Las métricas de desfase del consumidor cuantifican la diferencia entre los datos más recientes escritos sobre sus temas y los datos leídos por las aplicaciones. Amazon MSK proporciona las siguientes métricas de retraso en el consumo, que puede obtener a través de Amazon CloudWatch o mediante la supervisión abierta con Prometheus:,,, y. `EstimatedMaxTimeLag` `EstimatedTimeLag` `MaxOffsetLag` `OffsetLag` `SumOffsetLag` Para obtener información acerca de estas métricas, consulte the section called “CloudWatch métricas para corredores estándar”.

Amazon MSK admite las métricas de desfase del consumidor para clústeres con la versión 2.2.1 o posterior de Apache Kafka. Cuando trabaje con Kafka y con métricas, tenga en cuenta los siguientes puntos: CloudWatch

- Las métricas de retardo de consumo se emiten solo si un grupo de consumidores está en estado ESTABLE o VACÍO. Un grupo de consumidores se mantiene ESTABLE tras completar satisfactoriamente el proceso de reequilibrio, lo que garantiza que sus particiones se distribuyan uniformemente entre los consumidores.
- Las métricas de retraso de consumo están ausentes en los siguientes escenarios:
  - Si el grupo de consumidores es inestable.
  - El nombre del grupo de consumidores contiene dos puntos (:).
  - No has establecido la compensación por consumo para el grupo de consumidores.
- Los nombres de los grupos de consumidores se utilizan como dimensiones para las métricas de retraso de consumo CloudWatch. Si bien Kafka admite caracteres UTF-8 en los nombres de grupos de consumidores, solo CloudWatch admite caracteres ASCII para los valores de dimensión. Si utiliza caracteres que no son ASCII en los nombres de los grupos de consumidores, CloudWatch se eliminan las métricas de retardo de consumo. Para asegurarse de que las métricas

de retardo de consumo se reflejan correctamente CloudWatch, debe utilizar únicamente caracteres ASCII en los nombres de los grupos de consumidores.

Uso de las alertas de capacidad de almacenamiento de Amazon MSK

En los clústeres aprovisionados por Amazon MSK, usted elige la capacidad de almacenamiento principal del clúster. Si agota la capacidad de almacenamiento de un agente en el clúster aprovisionado, su capacidad de producir y consumir datos puede verse afectada, lo que provocará costosos tiempos de inactividad. Amazon MSK ofrece CloudWatch métricas que le ayudan a supervisar la capacidad de almacenamiento de su clúster. Sin embargo, para que le resulte más fácil detectar y resolver los problemas de capacidad de almacenamiento, Amazon MSK le envía automáticamente alertas dinámicas con respecto a la capacidad de almacenamiento de clústeres. Estas alertas incluyen recomendaciones sobre medidas a corto y largo plazo para administrar la capacidad de almacenamiento del clúster. Desde la consola de Amazon MSK, puede utilizar los enlaces rápidos de las alertas para tomar las medidas recomendadas de forma inmediata.

Existen dos tipos de alertas de capacidad de almacenamiento de MSK: preventivas y correctivas.

- Las alertas de capacidad de almacenamiento preventivas (“es necesario tomar medidas”) le advierten sobre posibles problemas de almacenamiento en su clúster. Cuando un agente de un clúster de MSK utilice más del 60 % o el 80 % de su capacidad de almacenamiento en disco, recibirá alertas preventivas sobre el agente afectado.
- Las alertas de capacidad de almacenamiento correctivas (“es necesario tomar medidas críticas”) le solicitan que tome medidas correctivas para solucionar un problema crítico del clúster cuando uno de los agentes de su clúster de MSK se queda sin capacidad de almacenamiento en disco.

Amazon MSK envía automáticamente estas alertas a la consola de Amazon MSK, AWS Health Dashboard EventBridge, Amazon y a los contactos de correo electrónico de su AWS cuenta. También puedes configurar Amazon EventBridge para que envíe estas alertas a Slack o a herramientas como New Relic y Datadog.

Las alertas con respecto a la capacidad de almacenamiento están habilitadas de manera predeterminada para todos los clústeres aprovisionados de MSK y no se pueden desactivar. Esta característica está disponible en todas las regiones en las que MSK está disponible.

Supervisión de las alertas de capacidad de almacenamiento

Puede comprobar las alertas de capacidad de almacenamiento de varias maneras:

- Vaya a la consola de Amazon MSK. Las alertas sobre la capacidad de almacenamiento se muestran en el panel de alertas del clúster durante 90 días. Incluyen recomendaciones y acciones mediante enlaces con un solo clic para abordar los problemas de capacidad de almacenamiento en disco.
- Usa ListClustersListClustersV2 o DescribeClusterV2 APIs para ver todas las alertas `CustomerActionStatus` de un clúster. DescribeCluster
- Vaya al panel de AWS Health para ver las alertas de MSK y otros servicios de AWS .
- Configura AWS Health API y Amazon EventBridge para redirigir las notificaciones de alertas a plataformas de terceros NewRelic, como Datadog y Slack.

Actualización de la configuración de seguridad de un clúster de Amazon MSK

Utilice la operación UpdateSecurityAmazon MSK para actualizar la configuración de autenticación y cifrado cliente-intermediario del clúster de MSK. También puede actualizar la autoridad de seguridad privada que se utiliza para firmar los certificados de autenticación TLS mutua. No puede cambiar la configuración de cifrado dentro del clúster (). broker-to-broker

El clúster debe tener el estado `ACTIVE` para que pueda actualizar la configuración de seguridad.

Si activa la autenticación mediante IAM, SASL o TLS, también debe activar el cifrado entre clientes y agentes. La tabla siguiente muestra las combinaciones posibles.

Autenticación	Opciones de cifrado entre el cliente y el agente	Cifrado entre agente y agente
Unauthenticated	TLS, PLAINTEXT, TLS_PLAINTEXT	Puede estar activado o desactivado.
mTLS	TLS, TLS_PLAINTEXT	Debe estar activado.
SASL/SCRAM	TLS	Debe estar activado.
SASL/IAM	TLS	Debe estar activado.

Cuando el cifrado entre cliente y agente está establecido en `TLS_PLAINTEXT` y la autenticación de cliente en `mTLS`, Amazon MSK crea dos tipos de oyentes a los que los clientes se conectan: uno

para que los clientes se conecten mediante la autenticación mTLS con cifrado TLS y otro para que los clientes se conecten sin autenticación ni cifrado (texto no cifrado).

Para más información sobre la configuración de seguridad, consulte the section called “Seguridad”.

Actualice la configuración de seguridad del clúster de Amazon MSK mediante AWS Management Console

1. ¿Iniciar sesión en la AWS Management Console consola Amazon MSK y abrirla desde https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. Elija el clúster de MSK que quiere actualizar.
3. En la sección Configuración de seguridad, elija Editar.
4. Elija la configuración de autenticación y cifrado que quiere para el clúster y, luego, elija Guardar cambios.

Actualización de la configuración de seguridad del clúster de Amazon MSK mediante AWS CLI

1. Cree un archivo JSON que contenga la configuración de cifrado que quiere que tenga el clúster. A continuación se muestra un ejemplo.

   > **Note**
   >
   > Solo puede actualizar la configuración de cifrado entre el cliente y el agente. No puede actualizar la configuración de cifrado integrada en el clúster (broker-to-broker).

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

2. Cree un archivo JSON que contenga la configuración de autenticación que quiere que tenga el clúster. A continuación se muestra un ejemplo.

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

3. Ejecute el siguiente AWS CLI comando:

```
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
```

El resultado de esta operación `update-security` se parece al siguiente JSON.

```
{

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

4. Para ver el estado de la `update-security` operación, ejecute el siguiente comando y *`ClusterOperationArn`* sustitúyalo por el ARN que obtuvo en el resultado del `update-security` comando.

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

El resultado de este comando `describe-cluster-operation` tendrá un aspecto similar al siguiente.

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

Si `OperationState` tiene el valor `PENDING` o `UPDATE_IN_PROGRESS`, espere un poco y vuelva a ejecutar el comando `describe-cluster-operation`.

> **Note**
>
> Las operaciones AWS CLI y las de la API para actualizar la configuración de seguridad de un clúster son idempotentes. Esto significa que si invoca la operación de actualización de seguridad y especifica una configuración de autenticación o cifrado que sea la misma que tiene el clúster actualmente, esa configuración no cambiará.

Actualización de la configuración de seguridad de un clúster mediante la API

Para actualizar la configuración de seguridad de un clúster de Amazon MSK mediante la API, consulte UpdateSecurity.

> **Note**
>
> Las operaciones de API AWS CLI y las de actualización de la configuración de seguridad de un clúster de MSK son idempotentes. Esto significa que si invoca la operación de actualización de seguridad y especifica una configuración de autenticación o cifrado que sea la misma que tiene el clúster actualmente, esa configuración no cambiará.

Expansión de la cantidad de agentes en un clúster de Amazon MSK

Utilice esta operación de Amazon MSK cuando quiera aumentar el número de agentes en el clúster de MSK. Para ampliar un clúster, asegúrese de que está en el estado `ACTIVE`.

> **Important**
>
> Si quiere expandir un clúster de MSK, asegúrese de utilizar esta operación de Amazon MSK. No intente agregar agentes a un clúster sin usar esta operación.

Para obtener información acerca de cómo volver a equilibrar particiones después de agregar agentes a un clúster, consulte the section called “Reasignar particiones”.

Amplíe un clúster de Amazon MSK mediante AWS Management Console

Este proceso describe cómo aumentar el número de agentes en un clúster de Amazon MSK mediante la AWS Management Console.

1. ¿Iniciar sesión en la AWS Management Console consola Amazon MSK y abrirla desde https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. Elija el clúster de MSK para aumentar su número de agentes.
3. En el menú desplegable Acciones, selecciona Editar número de corredores.
4. Introduzca el número de agentes que quiere que tenga el clúster por zona de disponibilidad y, luego, elija Guardar cambios.

Amplíe un clúster de Amazon MSK mediante AWS CLI

Este proceso describe cómo aumentar el número de agentes en un clúster de Amazon MSK mediante la AWS CLI.

1. Ejecute el siguiente comando, reemplazando *ClusterArn* por el nombre de recurso de Amazon (ARN) que obtuvo al crear el clúster. Si no tiene el ARN para su clúster, puede encontrarlo enumerando todos los clústeres. Para obtener más información, consulte the section called "Enumerar clústeres".

   Reemplace *Current-Cluster-Version* con la versión actual del clúster.

   > **Important**
   >
   > Las versiones de clúster no son enteros simples. Para buscar la versión actual del clúster, utilice la DescribeClusteroperación o el comando AWS CLI describe-cluster. Un ejemplo de ID de versión es `KTVPDKIKX0DER`.

   El *Target-Number-of-Brokers* parámetro representa el número total de nodos intermediarios que desea que tenga el clúster cuando la operación se complete correctamente. El valor que especifique *Target-Number-of-Brokers* debe ser un número entero superior al número actual de corredores del clúster. También debe ser un múltiplo del número de zonas de disponibilidad.

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

El resultado de esta operación `update-broker-count` se parece al siguiente JSON.

```
{

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

2. Para obtener el resultado de la `update-broker-count` operación, ejecute el siguiente comando y *`ClusterOperationArn`* reemplácelo por el ARN que obtuvo en el resultado del `update-broker-count` comando.

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

El resultado de este comando `describe-cluster-operation` tendrá un aspecto similar al siguiente.

```
{
    "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
        }
```

```
    }
}
```

En esta salida, `OperationType` es `INCREASE_BROKER_COUNT`. Si `OperationState` tiene el valor `UPDATE_IN_PROGRESS`, espere un rato y vuelva a ejecutar el comando `describe-cluster-operation`.

Expansión de un clúster de Amazon MSK mediante la API

Para aumentar el número de corredores de un clúster que utilizan la API, consulte UpdateBrokerCount.

Eliminación de un agente de un clúster de Amazon MSK

Utilice esta operación de Amazon MSK cuando quiera eliminar agentes de los clústeres aprovisionados por Amazon Managed Streaming for Apache Kafka (MSK). Puede reducir la capacidad de almacenamiento y cómputo de su clúster eliminando grupos de agentes, sin que ello afecte a la disponibilidad, ponga en riesgo la durabilidad de los datos ni interrumpa sus aplicaciones de flujo de datos.

Puede agregar más agentes a su clúster para gestionar el aumento del tráfico y eliminarlos cuando el tráfico disminuya. Con la capacidad de agregar y eliminar agentes, podrá utilizar mejor la capacidad de su clúster y optimizar los costos de infraestructura de MSK. La eliminación de un agente le permite controlar a nivel de agente la capacidad del clúster existente para adaptarse a sus necesidades de carga de trabajo y evitar la migración a otro clúster.

Utilice la AWS consola, la interfaz de línea de comandos (CLI), el SDK o AWS CloudFormation reduzca el número de agentes del clúster aprovisionado. MSK selecciona los agentes que no tienen particiones (excepto en el caso de temas canario) e impide que las aplicaciones generen datos para esos agentes, al tiempo que los elimina del clúster de forma segura.

Si quiere reducir el almacenamiento y la computación de un clúster, debe eliminar un agente por zona de disponibilidad. Por ejemplo, puede eliminar dos agentes de un clúster de dos zonas de disponibilidad o tres agentes de un clúster de tres zonas de disponibilidad en una sola operación de eliminación de agentes.

Para obtener información acerca de cómo volver a equilibrar particiones después de eliminar agentes de un clúster, consulte the section called “Reasignar particiones”.

Puede eliminar agentes de todos los clústeres de MSK aprovisionados basados en M5 y M7g, independientemente del tamaño de la instancia.

En las versiones 2.8.1 y posteriores de Kafka, incluso en los clústeres en modo, se puede eliminar el KRaft agente.

Temas

- Prepárese para eliminar los agentes quitando todas las particiones
- Elimine un corredor con la consola AWS de administración
- Eliminar un corredor con la AWS CLI
- Elimine un corredor con la AWS API

Prepárese para eliminar los agentes quitando todas las particiones

Antes de iniciar el proceso de eliminación de los agentes, mueva primero todas las particiones, excepto unas de los temas `__amazon_msk_canary` y `__amazon_msk_canary_state` de los agentes que planea eliminar. Se trata de temas internos que Amazon MSK crea para las métricas de estado y diagnóstico del clúster.

Puede usar Kafka admin APIs o Cruise Control para mover las particiones a otros corredores que desee conservar en el clúster. Consulte Reasignación de particiones.

Ejemplo de proceso de eliminación de particiones

Esta sección es un ejemplo de cómo eliminar las particiones del agente que desea eliminar. Suponga que tiene un clúster con 6 agentes, 2 agentes en cada zona de disponibilidad y que tiene cuatro temas:

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

1. Cree un equipo cliente tal y como se describe en Creación de un equipo cliente.
2. Tras configurar el equipo cliente, ejecute el siguiente comando para enumerar todos los temas disponibles en el clúster.

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

En este ejemplo, vemos cuatro nombres de temas: `__amazon_msk_canary`, `__consumer_offsets`, `__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-c657f7e4ff32-2` y `msk-brk-rmv`.

3. Cree un archivo json llamado `topics.json` en el equipo cliente y agregue todos los nombres de los temas de usuario, tal y como se muestra en el siguiente código de ejemplo. No es necesario incluir el nombre del tema `__amazon_msk_canary`, ya que se trata de un tema administrado por un servicio que se moverá automáticamente cuando sea necesario.

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

4. Ejecute el siguiente comando para generar una propuesta para mover las particiones a solo 3 agentes de los 6 agentes del clúster.

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

5. Cree un archivo llamado `reassignment-file.json` y copie el comando `proposed partition reassignment configuration` que obtuvo de arriba.
6. Ejecute el siguiente comando para mover las particiones que especificó en el `reassignment-file.json`.

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

El resultado tiene un aspecto similar al siguiente:

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

7. Ejecute el siguiente comando para verificar que todas las particiones se han movido.

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

   El resultado tiene un aspecto similar al siguiente. Supervise el estado hasta que todas las particiones de los temas solicitados se hayan reasignado correctamente:

```
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. Cuando el estado indique que se ha completado la reasignación de particiones para cada partición, supervise las métricas `UserPartitionExists` durante 5 minutos para asegurarse de que muestren `0` a los agentes desde los que ha movido las particiones. Tras confirmarlo, puede proceder a eliminar el agente del clúster.

Elimine un corredor con la consola AWS de administración

Para eliminar corredores con la consola de AWS administración

1. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/.
2. Elija el clúster de MSK que contiene los agentes que desea eliminar.
3. En la página de detalles del clúster, pulse el botón Acciones y seleccione la opción Editar número de agentes.
4. Introduzca el número de agentes que desea que el clúster tenga por zona de disponibilidad. La consola resume el número de agentes de todas las zonas de disponibilidad que se eliminarán. Asegúrese de que es lo que quiere.
5. Seleccione Save changes (Guardar cambios).

Para evitar la eliminación accidental de agentes, la consola le pide que confirme que desea eliminar agentes.

Eliminar un corredor con la AWS CLI

Ejecute el siguiente comando, reemplazando `ClusterArn`por el nombre de recurso de Amazon (ARN) que obtuvo al crear el clúster. Si no tiene el ARN para su clúster, puede encontrarlo enumerando todos los clústeres. Para obtener más información, consulte Enumeración de clústeres de Amazon MSK. Reemplace `Current-Cluster-Version` con la versión actual del clúster.

> **Important**
>
> Las versiones de clúster no son enteros simples. Para encontrar la versión actual del clúster, utilice la DescribeClusteroperación o el comando AWS CLI describe-cluster. Un ejemplo de ID de versión es `KTVPDKIKX0DER`.

El *`Target-Number-of-Brokers`* parámetro representa el número total de nodos intermediarios que desea que tenga el clúster cuando la operación se complete correctamente. El valor que especifique *`Target-Number-of-Brokers`* debe ser un número entero inferior al número actual de corredores del clúster. También debe ser un múltiplo del número de zonas de disponibilidad.

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

El resultado de esta operación `update-broker-count` se parece al siguiente JSON.

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

```
        }
    }
}
```

En esta salida, `OperationType` es `DECREASE_BROKER_COUNT`. Si `OperationState` tiene el valor `UPDATE_IN_PROGRESS`, espere un rato y vuelva a ejecutar el comando `describe-cluster-operation`.

Elimine un corredor con la AWS API

Para eliminar a los agentes de un clúster mediante la API, consulte la referencia UpdateBrokerCountde la API Amazon Managed Streaming for Apache Kafka.

Actualización del tamaño del agente del clúster de Amazon MSK

Puede escalar el clúster de MSK bajo demanda al cambiar el tamaño de los agentes sin tener que reasignar las particiones de Apache Kafka. Al cambiar el tamaño del agente, tendrá la flexibilidad necesaria para ajustar la capacidad de computación en clústeres de MSK en función de los cambios en las cargas de trabajo, sin interrumpir las E/S del clúster. Amazon MSK utiliza el mismo tamaño del agente para todos los agentes de un clúster determinado.

Esta sección describe cómo actualizar el tamaño del agente del clúster de MSK. En el caso de los corredores estándar, puede actualizar el tamaño del bróker del clúster de M5 o T3 a M7g, o de M7g a M5. En el caso de los corredores Express, solo puede utilizar corredores de tamaños m7g.

> **Note**
>
> No puede migrar de un tamaño de corredor más grande a uno más pequeño. Por ejemplo, de m7g.Large a T3.small.

Tenga en cuenta que la migración a un tamaño de agente más pequeño puede disminuir el rendimiento y reducir el rendimiento máximo alcanzable por agente. La migración a un bróker de mayor tamaño puede aumentar el rendimiento, pero puede costar más.

La actualización del tamaño del agente se hace de forma continua mientras el clúster está activo y en funcionamiento. Esto significa que Amazon MSK elimina un agente a la vez para actualizar el tamaño del agente. Para obtener información sobre cómo ofrecer alta disponibilidad en un clúster

durante la actualización del tamaño del agente, consulte the section called “Crear clústeres de alta disponibilidad”. Para reducir aún más cualquier posible impacto en la productividad, puede actualizar el tamaño del agente durante un periodo de poco tráfico.

Durante una actualización del tamaño del agente, puede seguir produciendo y consumiendo datos. Sin embargo, debe esperar a que finalice la actualización para poder reiniciar los agentes o invocar cualquiera de las operaciones de actualización que figuran en la sección Operaciones de Amazon MSK.

Si quiere actualizar el clúster a un tamaño del agente más pequeño, le recomendamos que intente primero actualizar un clúster de prueba para ver cómo afecta a su escenario.

> **Important**
>
> No se puede actualizar un clúster a un tamaño del agente más pequeño si el número de particiones por agente supera el número máximo especificado en the section called “ Ajuste el tamaño correcto del clúster: número de particiones por agente”.

Actualice el tamaño del bróker de clústeres de Amazon MSK mediante el AWS Management Console

Este proceso muestra cómo actualizar el tamaño del bróker de clústeres de Amazon MSK mediante el AWS Management Console

1. ¿Iniciar sesión en la AWS Management Console consola Amazon MSK y abrirla desde https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. Elija el clúster de MSK para el cual quiere actualizar el tamaño del agente.
3. En la página de detalles del clúster, encuentre la sección Resumen de agentes y elija Editar tamaño del agente.
4. Elija el tamaño del agente que quiere de la lista.
5. Guarde los cambios.

Actualice el tamaño del bróker de clústeres de Amazon MSK mediante el AWS CLI

Ejecute el siguiente comando, reemplazando *ClusterArn*por el nombre de recurso de Amazon (ARN) que obtuvo al crear el clúster. Si no tiene el ARN para su clúster, puede encontrarlo

enumerando todos los clústeres. Para obtener más información, consulte the section called "Enumerar clústeres".

1. *Current-Cluster-Version*Sustitúyalo por la versión actual del clúster y *TargetType* por el nuevo tamaño que desee que tengan los corredores. Para más información acerca de los tamaños del agente, consulte the section called "Tipos de agente".

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

   A continuación, se muestra un ejemplo de cómo utilizar este comando.

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

   El resultado de este comando tendrá un aspecto similar al siguiente.

   ```
   {
       "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. Para obtener el resultado de la `update-broker-type` operación, ejecute el siguiente comando y *ClusterOperationArn* reemplácelo por el ARN que obtuvo en el resultado del `update-broker-type` comando.

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

   El resultado de este comando `describe-cluster-operation` tendrá un aspecto similar al siguiente.

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

Si `OperationState` tiene el valor `UPDATE_IN_PROGRESS`, espere un rato y vuelva a ejecutar el comando `describe-cluster-operation`.

Actualización del tamaño del agente mediante la API

Para actualizar el tamaño del bróker mediante la API, consulte UpdateBrokerType.

Puede utilizar `UpdateBrokerType` para actualizar el tamaño del agente de clúster de M5 o T3 a M7g, o de M7g a M5.

Utilice LinkedIn el control de crucero para Apache Kafka con Amazon MSK

Puede utilizar el LinkedIn Cruise Control para reequilibrar el clúster de Amazon MSK, detectar y corregir anomalías y supervisar el estado y el estado del clúster.

Descarga y compilación de Cruise Control

1. Cree una EC2 instancia de Amazon en la misma Amazon VPC que el clúster de Amazon MSK.
2. Instala Prometheus en la instancia de EC2 Amazon que creaste en el paso anterior. Anote la IP privada y el puerto. El número de puerto predeterminado es 9090. Para obtener información acerca de cómo configurar Prometheus para agregar métricas para su clúster, consulte the section called “Supervisión con Prometheus”.
3. Descarga Cruise Control en la EC2 instancia de Amazon. (Como alternativa, si lo prefieres, puedes usar una EC2 instancia de Amazon independiente para Cruise Control). Si el clúster tiene una versión de Apache Kafka anterior a la 2.4.*, utilice la última versión 2.4.* de Cruise

Control. Si el clúster tiene una versión de Apache Kafka anterior a la 2.4.*, utilice la última versión 2.0.* de Cruise Control.

4. Descomprima el archivo de Cruise Control y, a continuación, vaya a la carpeta descomprimida.
5. Ejecute el siguiente comando para instalar git.

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

6. Ejecute el siguiente comando para inicializar el repositorio local. *Your-Cruise-Control-Folder*Sustitúyelo por el nombre de la carpeta actual (la carpeta que obtuviste al descomprimir la descarga de Cruise Control).

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

7. Ejecute el comando siguiente para instalar y compilar el código fuente.

```
./gradlew jar copyDependantLibs
```

Configuración y ejecución de Cruise Control

1. Lleve a cabo las siguientes actualizaciones en el archivo `config/cruisecontrol.properties`. Sustituya los servidores de arranque de ejemplo y la cadena bootstrap-brokers con los valores de su clúster. Para obtener estas cadenas para su clúster, puede ver los detalles del clúster en la consola. Como alternativa, puede usar las operaciones GetBootstrapBrokersy DescribeClusterAPI o sus equivalentes de CLI.

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

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

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

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

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

En el caso de los corredores exprés, le recomendamos que no los utilice `DiskCapacityGoal` en ninguno de los objetivos configurados en las configuraciones de su analizador.

2. Edite el archivo `config/capacityCores.json` para especificar el tamaño de disco y los núcleos de CPU correctos, así como los límites de entrada/salida de la red. En el caso de los corredores de Express, la entrada de `DISK` capacidad solo es necesaria para configurar el Cruise Control. Dado que MSK gestiona todo el almacenamiento de los corredores de Express, deberías establecer este valor en un número extremadamente alto, como `Integer.MAX_VALUE` (`2147483647`) por ejemplo. Para los corredores estándar, puede usar la operación DescribeClusterAPI (o la CLI describe-cluster) para obtener el tamaño del disco. Para conocer los núcleos de CPU y los límites de entrada/salida de la red, consulte Tipos de EC2 instancias de Amazon.

   Standard broker config/capacityCores.json

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

Express broker config/capacityCores.json

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

3. Si lo desea, puede instalar la interfaz de usuario de Cruise Control. Para descargarla, vaya a Setting Up Cruise Control Frontend.
4. Ejecute el siguiente comando para iniciar Cruise Control. Considere usar una herramienta como `screen` o `tmux` para mantener abierta una sesión de larga duración.

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

5. Usa el Cruise Control APIs o la interfaz de usuario para asegurarte de que Cruise Control dispone de los datos de carga del clúster y de que hace sugerencias para reequilibrarlo. Podría llevarle varios minutos a obtener una ventana de métricas válida.

> **Important**
>
> Solo las versiones 2.5.60 y posteriores de Cruise Control son compatibles con los corredores Express, ya que los corredores Express no exponen los puntos finales de Zookeeper.

Utilización de la plantilla de implementación automatizada de Cruise Control para Amazon MSK

También puede utilizar esta CloudFormation plantilla para implementar fácilmente Cruise Control y Prometheus para obtener información más detallada sobre el rendimiento de su clúster de Amazon MSK y optimizar la utilización de los recursos.

Características clave:

- Aprovisionamiento automatizado de una EC2 instancia de Amazon con Cruise Control y Prometheus preconfigurados.
- Compatibilidad con el clúster aprovisionado de Amazon MSK.
- Autenticación flexible con un IAM. PlainText
- Cruise Control no depende de Zookeeper.
- Proporcione sus propios archivos de configuración almacenados en un bucket de Amazon S3 para personalizar fácilmente los destinos de Prometheus, los ajustes de capacidad de Cruise Control y otras configuraciones.

Guía de reequilibrio de particiones

Directrices para la reasignación de particiones de Kafka

La reasignación de particiones en Kafka puede requerir muchos recursos, ya que implica la transferencia de datos importantes entre intermediarios, lo que puede provocar congestión en la red y afectar a las operaciones de los clientes. Las siguientes prácticas recomendadas le ayudan a gestionar la reasignación de particiones de forma eficaz mediante el ajuste de las velocidades de aceleración, el aprovechamiento de los controles de simultaneidad y la comprensión de los tipos de reasignación para minimizar las interrupciones en las operaciones del clúster.

Cómo gestionar la simultaneidad en Cruise Control

El control de crucero proporciona parámetros de ajuste automático para controlar la simultaneidad de los movimientos de partición y liderazgo. Los siguientes parámetros ayudan a mantener una carga aceptable durante las reasignaciones:

- Movimiento máximo de particiones simultáneas: defina la opción **`num.concurrent.partition.movements.per.broker`** para limitar los movimientos simultáneos de particiones entre intermediarios, evitando un uso excesivo de la red.

Example Ejemplo

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

Esta configuración limita a cada corredor a mover no más de 10 particiones en un momento dado, lo que equilibra la carga entre los corredores.

Utilice la regulación para controlar el ancho de banda

- Parámetro de aceleración: Al realizar una reasignación de particiones con`kafka-reassign-partitions.sh`, utilice el `--throttle parameter` para establecer una velocidad de transferencia máxima (en bytes por segundo) para el movimiento de datos entre corredores.

  Example Ejemplo

  ```
  --throttle 5000000
  ```

  Esto establece un ancho de banda máximo de 5 MB/s.
- Equilibre los ajustes del acelerador: es crucial elegir una velocidad de aceleración adecuada:

  Si se ajusta a un nivel demasiado bajo, la reasignación puede tardar mucho más.

  Si se establece un valor demasiado alto, es posible que los clientes experimenten un aumento de la latencia.
- Comience con una velocidad de aceleración conservadora y ajústela en función de la supervisión del rendimiento del clúster. Pruebe el acelerador que elija antes de aplicarlo a un entorno de producción para encontrar el equilibrio óptimo.

Pruebe y valide en un entorno de ensayo

Antes de implementar las reasignaciones en la producción, realice las pruebas de carga en un entorno de ensayo con configuraciones similares. Esto le permite ajustar los parámetros y minimizar los impactos inesperados en la producción en vivo.

Actualización de la configuración de un clúster de Amazon MSK

Para actualizar la configuración de un clúster, asegúrese de que el clúster está en el estado `ACTIVE`. También debe asegurarse de que el número de particiones por agente en el clúster de MSK esté por debajo de los límites descritos en the section called " Ajuste el tamaño correcto del clúster: número de particiones por agente". No puede actualizar la configuración de un clúster que supere estos límites.

Para obtener información acerca de la configuración de MSK, incluido cómo crear una configuración personalizada, qué propiedades puede actualizar y qué sucede al actualizar la configuración de un clúster existente, consulte the section called "Configuración de bróker".

Temas

- Disponibilidad del bróker durante las actualizaciones de configuración
- Actualización de la configuración de un clúster mediante el AWS CLI
- Actualización de la configuración de un clúster de Amazon MSK mediante la API

Disponibilidad del bróker durante las actualizaciones de configuración

Amazon MSK mantiene una alta disponibilidad durante la mayoría de las actualizaciones de configuración de clústeres. Amazon MSK realiza una actualización progresiva en la que actualiza un bróker a la vez. Durante este proceso, el clúster permanece disponible, aunque los corredores individuales se reiniciarán a medida que se actualicen sus configuraciones. Sin embargo, algunos cambios de configuración pueden requerir que todos los corredores se actualicen simultáneamente, lo que puede provocar una breve interrupción en todo el clúster. Para obtener más información sobre el impacto en la disponibilidad de los corredores durante las actualizaciones, consulte. Configuración aprovisionada de Amazon MSK

Antes de actualizar los clústeres de producción, le recomendamos que pruebe los cambios de configuración en un entorno que no sea de producción y que programe las actualizaciones durante los períodos de mantenimiento.

Si tiene algún problema al actualizar su clúster de MSK, consulte ¿Cómo puedo solucionar problemas al actualizar mi clúster de Amazon MSK?

Actualización de la configuración de un clúster mediante el AWS CLI

1. Copie el siguiente JSON y guárdelo en un archivo. Nombre el archivo `configuration-info.json`. *`ConfigurationArn`*Sustitúyalo por el nombre de recurso de Amazon (ARN) de la configuración que quieres usar para actualizar el clúster. La cadena ARN debe estar entre comillas en el siguiente JSON.

   *`Configuration-Revision`*Sustitúyalo por la revisión de la configuración que desee usar. Las revisiones de configuración son enteros (números enteros) que comienzan por 1. Este entero no debe estar entre comillas en el siguiente JSON.

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

2. Ejecute el siguiente comando y *`ClusterArn`* reemplácelo por el ARN que obtuvo al crear el clúster. Si no tiene el ARN para su clúster, puede encontrarlo enumerando todos los clústeres. Para obtener más información, consulte the section called “Enumerar clústeres”.

   *`Path-to-Config-Info-File`*Sustitúyalo por la ruta al archivo de información de configuración. Si asignó un nombre al archivo que creó en el paso anterior `configuration-info.json` y lo guardó en el directorio actual, entonces *`Path-to-Config-Info-File`* es`configuration-info.json`.

   Reemplace *`Current-Cluster-Version`* con la versión actual del clúster.

   > **Important**
   >
   > Las versiones de clúster no son enteros simples. Para buscar la versión actual del clúster, utilice la DescribeClusteroperación o el comando AWS CLI describe-cluster. Un ejemplo de ID de versión es `KTVPDKIKX0DER`.

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

   A continuación, se muestra un ejemplo de cómo utilizar este comando.

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

El resultado de este comando `update-cluster-configuration` tendrá un aspecto similar al siguiente.

```
{
    "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. Para obtener el resultado de la `update-cluster-configuration` operación, ejecute el siguiente comando y *`ClusterOperationArn`* reemplácelo por el ARN que obtuvo en el resultado del `update-cluster-configuration` comando.

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

El resultado de este comando `describe-cluster-operation` tendrá un aspecto similar al siguiente.

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

En esta salida, `OperationType` es `UPDATE_CLUSTER_CONFIGURATION`. Si `OperationState` tiene el valor `UPDATE_IN_PROGRESS`, espere un rato y vuelva a ejecutar el comando `describe-cluster-operation`.

Actualización de la configuración de un clúster de Amazon MSK mediante la API

Para usar la API para actualizar la configuración de un clúster de Amazon MSK, consulte UpdateClusterConfiguration.

Reinicio de un agente para un clúster de Amazon MSK

Utilice esta operación de Amazon MSK cuando quiera reiniciar el agente de un clúster de MSK. Para reiniciar un agente de un clúster, asegúrese de que el clúster esté en estado `ACTIVE`.

Es posible que el servicio Amazon MSK reinicie los agentes de su clúster de MSK durante el mantenimiento del sistema, por ejemplo, al aplicar los parches o actualizar las versiones. Reiniciar un agente manualmente le permite probar la resistencia de sus clientes de Kafka para determinar cómo responden al mantenimiento del sistema.

Reinicie un bróker para un clúster de Amazon MSK mediante el AWS Management Console

Este proceso describe cómo reiniciar un agente para un clúster de Amazon MSK mediante. AWS Management Console

1. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/.
2. Elija el clúster de MSK para reiniciar su agente.
3. Desplácese hacia abajo hasta la sección Detalles del agente y elija el agente que desee reiniciar.
4. Pulse el botón Reiniciar el agente.

Reinicie un bróker para un clúster de Amazon MSK mediante el AWS CLI

Este proceso describe cómo reiniciar un agente para un clúster de Amazon MSK mediante. AWS CLI

1. Ejecute el siguiente comando y *ClusterArn* sustitúyalo por el nombre de recurso de Amazon (ARN) que obtuvo al crear el *BrokerId* clúster y por el ID del agente que desea reiniciar.

   > **Note**
   >
   > La operación `reboot-broker` solo permite reiniciar un agente a la vez.

   Si no tiene el ARN para su clúster, puede encontrarlo enumerando todos los clústeres. Para obtener más información, consulte the section called “Enumerar clústeres”.

   Si no tiene el intermediario IDs de su clúster, puede encontrarlo enumerando los nodos del intermediario. Para obtener más información sobre los nodos, consulte list-nodes.

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

   El resultado de esta operación `reboot-broker` se parece al siguiente JSON.

   ```
   {

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

2. Para obtener el resultado de la `reboot-broker` operación, ejecute el siguiente comando y *ClusterOperationArn* reemplácelo por el ARN que obtuvo en el resultado del `reboot-broker` comando.

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

   El resultado de este comando `describe-cluster-operation` tendrá un aspecto similar al siguiente.

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

Cuando se complete la operación de reinicio, `OperationState` será `REBOOT_COMPLETE`.

Reinicio de un clúster de Amazon MSK mediante la API

Para reiniciar un bróker de un clúster mediante la API, consulte RebootBroker.

Etiquetado de un clúster de Amazon MSK

Puede asignar sus propios metadatos en forma de etiquetas a un recurso de Amazon MSK, como un clúster de MSK. Una etiqueta es un par clave-valor definido por el usuario para un recurso. El uso de etiquetas es una forma sencilla pero eficaz de gestionar AWS los recursos y organizar los datos, incluidos los datos de facturación.

Temas

- Conceptos básicos sobre etiquetas para clústeres de Amazon MSK
- Realice un seguimiento de los costos de los clústeres de Amazon MSK mediante el etiquetado
- Restricciones de las etiquetas
- Etiquetado de recursos mediante la API de Amazon MSK

Conceptos básicos sobre etiquetas para clústeres de Amazon MSK

Puede utilizar la API de Amazon MSK para finalizar las siguientes tareas:

- Agregar etiquetas a un recurso de Amazon MSK.
- Mostrar las etiquetas de un recurso de Amazon MSK.
- Quitar las etiquetas de un recurso de Amazon MSK.

Puede utilizar las etiquetas para categorizar los recursos de Amazon MSK. Por ejemplo, puede categorizar los clústeres de Amazon MSK según su finalidad, propietario o entorno. Dado que define la clave y el valor de cada etiqueta, puede crear un conjunto de categorías personalizadas para satisfacer sus necesidades específicas. Por ejemplo, podría definir un conjunto de etiquetas que le ayude a realizar un seguimiento de los clústeres por propietario y aplicaciones asociadas.

A continuación, se muestran varios ejemplos de etiquetas:

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

Realice un seguimiento de los costos de los clústeres de Amazon MSK mediante el etiquetado

Puede usar etiquetas para categorizar y realizar un seguimiento de sus AWS costos. Cuando aplica etiquetas a sus AWS recursos, incluidos los clústeres de Amazon MSK, el informe de asignación de AWS costes incluye el uso y los costes agregados por etiquetas. Si aplica etiquetas que representen categorías de negocio (por ejemplo, centros de costos, nombres de aplicaciones o propietarios), puede organizar los costos entre diferentes servicios. Para obtener más información, consulte Utilizar etiquetas de asignación de costos para informes de facturación personalizados en la Guía del usuario de AWS Billing .

Restricciones de las etiquetas

Se aplican las siguientes restricciones a las etiquetas en Amazon MSK.

Restricciones básicas

- El número máximo de etiquetas por recurso es 50.
- Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas.
- No se pueden cambiar ni editar etiquetas de un recurso eliminado.

Restricciones de clave de etiqueta

- Cada clave de etiqueta debe ser única. Si agrega una etiqueta con una clave que ya está en uso, la nueva etiqueta sobrescribe el par clave-valor existente.
- Una clave de etiqueta no puede comenzar por `aws:` porque este prefijo está reservado para su utilización por AWS. AWS crea etiquetas cuyo nombre comienza por este prefijo por usted, pero usted no puede editarlas ni eliminarlas.
- Las claves de etiqueta deben tener entre 1 y 128 caracteres Unicode de longitud.
- Las claves de etiquetas deben constar de los siguientes caracteres: letras Unicode, números, espacios en blanco y los siguientes caracteres especiales: `_ . / = + - @`.

Restricciones de valor de etiqueta

- Los valores de etiqueta deben tener entre 0 y 255 caracteres Unicode de longitud.
- Los valores de etiqueta pueden estar en blanco. De lo contrario, deben constar de los siguientes caracteres: letras Unicode, números, espacios en blanco y cualquiera de los siguientes caracteres especiales: `_ . / = + - @`.

Etiquetado de recursos mediante la API de Amazon MSK

Puede utilizar las siguientes operaciones para etiquetar o desetiquetar un recurso de Amazon MSK o para mostrar el conjunto actual de etiquetas de un recurso:

- ListTagsForResource
- TagResource
- UntagResource

Migración a un clúster de Amazon MSK

El Replicador Amazon MSK se puede utilizar para la migración de clústeres de MSK. Consulte ¿Qué es el Replicador Amazon MSK?. Como alternativa, puede usar Apache MirrorMaker 2.0 para migrar de un clúster que no sea de MSK a uno de Amazon MSK. Para ver un ejemplo de cómo hacerlo, consulte el artículo sobre migración de un clúster de Apache Kafka en las instalaciones a Amazon MSK mediante. MirrorMaker Para obtener información sobre cómo usarlo MirrorMaker, consulte Reflejar datos entre clústeres en la documentación de Apache Kafka. Se recomienda configurarlo MirrorMaker en una configuración de alta disponibilidad.

Un esquema de los pasos a seguir al utilizarlos MirrorMaker para migrar a un clúster de MSK

1. Cree el clúster de MSK de destino
2. Comience MirrorMaker desde una EC2 instancia de Amazon VPC en la misma VPC de Amazon VPC que el clúster de destino.
3. Inspeccione el MirrorMaker retraso.
4. Cuando se MirrorMaker ponga al día, redirija a los productores y consumidores al nuevo clúster utilizando los intermediarios bootstrap del clúster de MSK.
5. Cierre de MirrorMaker RRRRR

Migración de su clúster de Apache Kafka a Amazon MSK

Supongamos que tiene un clúster de Apache Kafka llamado `CLUSTER_ONPREM`. Dicho clúster se rellena con temas y datos. Si desea migrar dicho clúster a un nuevo clúster de Amazon MSK llamado `CLUSTER_AWSMSK`, este procedimiento ofrece una gran perspectiva de los pasos que debe seguir.

Migración de su clúster de Apache Kafka existente a Amazon MSK

1. En `CLUSTER_AWSMSK`, cree todos los temas que desee migrar.

   No puede utilizar MirrorMaker este paso porque no vuelve a crear automáticamente los temas que desea migrar con el nivel de replicación adecuado. Puede crear los temas en Amazon MSK con los mismos factores de replicación y números de particiones que tuviesen en `CLUSTER_ONPREM`. También puede crear los temas con distintos factores de replicación y números de particiones.
2. Comience MirrorMaker desde una instancia que tenga acceso de lectura `CLUSTER_ONPREM` y acceso de escritura. `CLUSTER_AWSMSK`
3. Ejecute el siguiente comando para duplicar todos los temas:

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

   En este comando, `config/mirrormaker-consumer.properties` señala a un agente de arranque en `CLUSTER_ONPREM`, por ejemplo, `bootstrap.servers=localhost:9092`. Y `config/mirrormaker-producer.properties` apunta a

un agente de arranque en CLUSTER_AWSMSK; por ejemplo,. `bootstrap.servers=10.0.0.237:9092,10.0.2.196:9092,10.0.1.233:9092`

4. Siga MirrorMaker ejecutándose en segundo plano y continúe usándolo. `CLUSTER_ONPREM` MirrorMaker refleja todos los datos nuevos.
5. Compruebe el progreso de la duplicación inspeccionando el intervalo entre el último desplazamiento de cada tema y el desplazamiento actual desde el que se MirrorMaker consume.

   Recuerde que MirrorMaker se trata simplemente de utilizar un consumidor y un productor. Por lo tanto, puede comprobar el intervalo utilizando la herramienta `kafka-consumer-groups.sh`. Para localizar el nombre del grupo de consumidores, mire en el archivo `mirrormaker-consumer.properties` para el `group.id` y utilice su valor. Si en el archivo no se encuentra dicha clave, puede crearla. Por ejemplo, establezca `group.id=mirrormaker-consumer-group`.
6. Cuando MirrorMaker termine de reflejar todos los temas, detenga a todos los productores y consumidores y, a continuación, deje de hacerlo. MirrorMaker A continuación, redirija a los productores y los consumidores al clúster de `CLUSTER_AWSMSK` cargando los valores de los agentes de arranque de sus productores y consumidores. Reinicie todos los productores y consumidores en `CLUSTER_AWSMSK`.

Migración de un clúster de Amazon MSK a otro

Puede usar Apache MirrorMaker 2.0 para migrar de un clúster que no sea de MSK a uno de Amazon MSK. Por ejemplo, puede migrar de una versión de Apache Kafka a otra. Para ver un ejemplo de cómo hacerlo, consulte el artículo sobre migración de un clúster de Apache Kafka en las instalaciones a Amazon MSK mediante. MirrorMaker De forma alternativa, el Replicador Amazon MSK se puede utilizar para la migración de clústeres de MSK. Para obtener más información acerca del Replicador Amazon MSK, consulte ¿Qué es el Replicador Amazon MSK?.

MirrorMaker Prácticas recomendadas de la versión 1.0

Esta lista de mejores prácticas se aplica a la MirrorMaker versión 1.0.

- Ejecute MirrorMaker en el clúster de destino. De esta forma, si se produce un problema de red, los mensajes seguirán estando disponibles en el clúster de origen. Si se ejecuta MirrorMaker en el clúster de origen y los eventos están almacenados en un búfer en el productor y hay un problema de red, es posible que se pierdan los eventos.
- Si en el tránsito es necesario cifrar, hágalo en el clúster de origen.

- Para los consumidores, establezca auto.commit.enabled=false
- Para los productores, establezca
  - max.in.flight.requests.per.connection=1
  - retries=Int.Max_Value
  - acks=all
  - max.block.ms = Long.Max_Value
- Para un alto rendimiento del productor:
  - Almacene mensajes en búfer y rellene lotes de mensajes: tune buffer.memory, batch.size, linger.ms
  - Ajuste los búferes de conectores: receive.buffer.bytes, send.buffer.bytes
- Para evitar la pérdida de datos, desactive la confirmación automática en el origen para que MirrorMaker pueda controlar las confirmaciones, lo que suele hacer después de recibir el paquete del clúster de destino. Si el productor tiene acks=all y el clúster de destino tiene min.insync.replicas establecido en más de 1, los mensajes se conservan en más de un intermediario en el destino antes de que el consumidor confirme la compensación en el origen. MirrorMaker
- Si el orden es importante, puede establecer los reintentos en 0. De manera alternativa, para un entorno de producción, establezca las conexiones en tránsito máximas en 1 para garantizar que los lotes enviados no se envían fuera del orden si un lote produce un error a la mitad. De esta forma, cada lote enviado se vuelve a intentar hasta que el siguiente lote se envía. Si max.block.ms no se establece al valor mínimo y el búfer del producto está completo, puede haber pérdida de datos (dependiendo de otras configuraciones). Esto puede bloquear y ejercer presión contra el consumidor.
- Para un gran rendimiento
  - Aumente la memoria del búfer.
  - Aumente el tamaño del lote.
  - Sincronice linger.ms para permitir que los lotes se completen. Esto también permite una mejor compresión, menos uso de la banda ancha de red y menos almacenamiento en el clúster. Todo esto resultar en una mayor retención.
  - Supervisar el uso de la memoria y la CPU.
- Para un gran rendimiento del consumidor
  - Aumente la cantidad de subprocesos/consumidores por MirrorMaker proceso: num.streams.
  - Aumente primero la cantidad de MirrorMaker procesos en las máquinas antes de aumentar los subprocesos para permitir una alta disponibilidad.

  - Aumente el número de MirrorMaker procesos primero en la misma máquina y, después, en máquinas diferentes (con el mismo ID de grupo).
  - Aísle los temas que tengan un rendimiento muy alto y utilice MirrorMaker instancias independientes.
- Para la administración y la configuración
  - Utilice AWS CloudFormation y configure herramientas de administración como Chef y Ansible.
  - Utilice montajes de Amazon EFS para mantener accesibles todos los archivos de configuración desde todas las EC2 instancias de Amazon.
  - Utilice contenedores para facilitar el escalado y la administración de las MirrorMaker instancias.
- Por lo general, se necesita más de un consumidor para saturar a un productor. MirrorMaker Por lo tanto, configure varios consumidores. En primer lugar, configúrelos en diferentes equipos para proporcionar alta disponibilidad. A continuación, escale equipos individuales hasta tener un consumidor para cada partición, con los consumidores distribuidos equitativamente entre los equipos.
- Para la adquisición y entrega de alto rendimiento, ajuste los búferes de recepción y envío, porque sus valores predeterminados podrían ser demasiado bajos. Para obtener el máximo rendimiento, asegúrese de que el número total de transmisiones (num.streams) coincida con todas las particiones temáticas que MirrorMaker se están intentando copiar al clúster de destino.

Ventajas de 2. MirrorMaker *

- Hace uso del marco y ecosistema de Apache Kafka Connect.
- Detecta nuevos temas y particiones.
- Sincroniza automáticamente la configuración de temas entre clústeres.
- Admite pares de clústeres «activo/activo», así como cualquier número de clústeres activos.
- Proporciona nuevas métricas, incluida end-to-end la latencia de replicación en varios centros de datos y clústeres.
- Emite las compensaciones necesarias para migrar consumidores entre clústeres y proporciona herramientas para traducir compensaciones.
- Admite un archivo de configuración de alto nivel para especificar varios clústeres y flujos de replicación en un solo lugar, en comparación con las propiedades de productor/consumidor de bajo nivel de cada MirrorMaker proceso 1.*.

Eliminar un clúster aprovisionado de Amazon MSK

> **Note**
>
> Si el clúster de Amazon MSK aprovisionado tiene una política de escalado automático, le recomendamos que la elimine antes de eliminar el clúster. Para obtener más información, consulte Escalado automático para clústeres de Amazon MSK.

Temas

- Elimine un clúster aprovisionado de Amazon MSK mediante AWS Management Console
- Elimine un clúster aprovisionado de Amazon MSK mediante AWS CLI
- Eliminar un clúster aprovisionado de Amazon MSK mediante la API

Elimine un clúster aprovisionado de Amazon MSK mediante AWS Management Console

Este proceso describe cómo eliminar un clúster aprovisionado de Amazon MSK mediante. AWS Management Console Antes de eliminar un clúster de MSK, asegúrese de tener una copia de seguridad de todos los datos importantes almacenados en el clúster y de que no haya ninguna tarea programada que dependa del clúster. No puede deshacer la eliminación de un clúster de MSK.

1. ¿Iniciar sesión en la AWS Management Console consola Amazon MSK y abrirla desde https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. Para elegir el clúster de MSK que quiere eliminar, marque la casilla situada junto a él.
3. Elija Eliminar y confirme la eliminación.

Elimine un clúster aprovisionado de Amazon MSK mediante AWS CLI

Este proceso describe cómo eliminar un clúster aprovisionado de MSK mediante. AWS CLI Antes de eliminar un clúster de MSK, asegúrese de tener una copia de seguridad de todos los datos importantes almacenados en el clúster y de que no haya ninguna tarea programada que dependa del clúster. No puede deshacer la eliminación de un clúster de MSK.

Ejecute el siguiente comando, reemplazando *ClusterArn* por el nombre de recurso de Amazon (ARN) que obtuvo al crear el clúster. Si no tiene el ARN para su clúster, puede encontrarlo

enumerando todos los clústeres. Para obtener más información, consulte the section called "Enumerar clústeres".

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

Eliminar un clúster aprovisionado de Amazon MSK mediante la API

La API de Amazon MSK le permite crear y gestionar mediante programación su clúster aprovisionado de MSK como parte de scripts de implementación o aprovisionamiento de infraestructura automatizados. Este proceso describe cómo eliminar un clúster aprovisionado de Amazon MSK mediante la API de Amazon MSK. Antes de eliminar un clúster de Amazon MSK, asegúrese de tener una copia de seguridad de todos los datos importantes almacenados en el clúster y de que no haya tareas programadas que dependan del clúster. No puede deshacer la eliminación de un clúster de MSK.

Para eliminar un clúster mediante la API, consulte. DeleteCluster

Características y conceptos clave de Amazon MSK

Los clústeres aprovisionados de Amazon MSK ofrecen una amplia gama de características y capacidades que le ayudan a optimizar el rendimiento del clúster y a satisfacer sus necesidades de streaming. En los temas siguientes se describen estas funcionalidades en detalle.

- la AWS Management Console,
- La referencia de la API de Amazon MSK
- La referencia de los comandos de la CLI de Amazon MSK

Temas

- Tipos de brókeres de Amazon MSK
- Tamaños del agente de Amazon MSK
- Gestión del almacenamiento para corredores estándar
- Seguridad en Amazon MSK
- Configuración aprovisionada de Amazon MSK
- Aplicación de parches
- El agente no está en línea y el cliente realiza una conmutación por error

- Registro de Amazon MSK
- Administración de metadatos
- Recursos de Amazon MSK
- Versiones de Apache Kafka
- Solución de problemas del clúster de Amazon MSK

Tipos de brókeres de Amazon MSK

MSK Provisioned ofrece dos tipos de corredores: Standard y Express. Los corredores estándar le ofrecen la mayor flexibilidad para configurar sus clústeres, mientras que los corredores Express ofrecen más elasticidad, rendimiento y capacidad de recuperación y permiten ejecutar aplicaciones de ease-of-use streaming de alto rendimiento. Consulte las siguientes subsecciones para obtener más información sobre cada oferta. La siguiente tabla también destaca la comparación de características clave entre los corredores Standard y Express.

Comparación de tipos de corredores aprovisionados por MSK

Característica	Bróker estándar	Broker exprés
Administración del almacenamiento	Gestionado por el cliente (las características incluyen el almacenamiento EBS, el almacenamiento por niveles, el rendimiento del almacenamiento aprovisionado, el escalado automático y las alertas de capacidad de almacenamiento)	Totalmente gestionado por MSK
Instancias compatibles	T3, M5, M7g	M7g
Consideraciones sobre el tamaño y la escala	Rendimiento, conexiones, particiones y almacenamiento	Rendimiento, conexiones, particiones
Escalamiento de Broker	Escalado vertical y horizontal	Escalado vertical y horizontal
Versiones de Kafka	Consulte Versiones de Apache Kafka	Comienza en la versión 3.6

Característica	Bróker estándar	Broker exprés
Configuración de Apache Kafka	Más configurable	Principalmente gestionado con MSK para una mayor resilienc ia
Seguridad	Cifrado, Private/Public acceso, autenticación y autorización: IAM, SASL/SCRAM, mTLS, texto plano, Kafka ACLs	Cifrado, Private/Public acceso, autenticación y autorización: IAM, SASL/SCRAM, mTLS, texto plano, Kafka ACLs
Supervisión	CloudWatch, Supervisión abierta	CloudWatch, Supervisión abierta

> **Note**
>
> No puede cambiar un clúster aprovisionado por MSK de un tipo de corredor estándar a uno de tipo Express cambiando el tipo de corredor mediante la API de MSK. Debe crear un clúster nuevo con el tipo de agente deseado (Standard o Express).

Temas

- Agentes estándar de Amazon MSK
- Agentes de Amazon MSK Express

Agentes estándar de Amazon MSK

Los agentes estándar de MSK Provisioned ofrecen la mayor flexibilidad para configurar el rendimiento del clúster. Puede elegir entre una amplia gama de configuraciones de clústeres para lograr las características de disponibilidad, durabilidad, rendimiento y latencia necesarias para sus aplicaciones. También puede aprovisionar la capacidad de almacenamiento y aumentarla según sea necesario. Amazon MSK se encarga del mantenimiento del hardware de los intermediarios estándar y los recursos de almacenamiento adjuntos, reparando automáticamente los problemas de hardware que puedan surgir. Encontrará más información en este documento sobre diversos temas relacionados con los intermediarios estándar, incluidos los temas relacionados con la administración, las configuraciones y el mantenimiento del almacenamiento.

Agentes de Amazon MSK Express

Los corredores Express para MSK Provisioned hacen que Apache Kafka sea más fácil de administrar, más rentable de ejecutar a escala y más elástico con la baja latencia esperada. Los intermediarios incluyen un pay-as-you-go almacenamiento que se escala automáticamente y no requiere dimensionamiento, aprovisionamiento ni supervisión proactiva. Según el tamaño de la instancia seleccionada, cada nodo de intermediario puede ofrecer hasta 3 veces más rendimiento por agente, ampliarse hasta 20 veces más rápido y realizar una recuperación un 90% más rápida en comparación con los brókers estándar de Apache Kafka. Los corredores Express vienen preconfigurados con las prácticas recomendadas por defecto de Amazon MSK y aplican cuotas de rendimiento de los clientes para minimizar la contención de recursos entre los clientes y las operaciones en segundo plano de Kafka.

Estos son algunos de los factores y capacidades clave que se deben tener en cuenta al utilizar los corredores de Express.

- Sin administración del almacenamiento: Express Brokers elimina la necesidad de aprovisionar o administrar cualquier recurso de almacenamiento. Obtiene un almacenamiento elástico pay-as-you-go, prácticamente ilimitado y totalmente administrado. En el caso de los casos de uso de alto rendimiento, no es necesario preocuparse por las interacciones entre las instancias de procesamiento y los volúmenes de almacenamiento, ni por los cuellos de botella relacionados con el rendimiento. Estas capacidades simplifican la administración de clústeres y eliminan la sobrecarga operativa de la administración del almacenamiento.
- Escalamiento más rápido: los corredores Express le permiten escalar su clúster y mover las particiones hasta 20 veces más rápido que con los corredores estándar. Esta capacidad es crucial cuando necesita escalar su clúster para hacer frente a los próximos picos de carga o escalar su clúster para reducir los costos. Consulte las secciones sobre la expansión del clúster, la eliminación de intermediarios, la reasignación de particiones y la configuración LinkedIn del Cruise Control para el reequilibrio para obtener más información sobre cómo escalar el clúster.
- Mayor rendimiento: los corredores Express ofrecen hasta 3 veces más rendimiento por corredor que los corredores estándar. Por ejemplo, puede escribir datos de forma segura hasta 500 MBps con cada bróker Express del tamaño de 7 mg.16 x, frente a los 153,8 con el bróker estándar equivalente (ambos números suponen una asignación de ancho de banda suficiente para las operaciones MBps en segundo plano, como la replicación y el reequilibrio).
- Configurados para ofrecer una alta resiliencia: los corredores Express ofrecen automáticamente varias prácticas recomendadas para mejorar la resiliencia de su clúster. Estas incluyen barreras para las configuraciones críticas de Apache Kafka, las cuotas de rendimiento y las reservas de

capacidad para operaciones en segundo plano y reparaciones no planificadas. Estas capacidades hacen que sea más seguro y fácil ejecutar aplicaciones de Apache Kafka a gran escala. Consulte las secciones sobre Configuraciones de Express Broker y Cuota de bróker de Amazon MSK Express para obtener más información.

- Sin ventanas de mantenimiento: No hay ventanas de mantenimiento para los corredores de Express. Amazon MSK actualiza automáticamente el hardware del clúster de forma continua. Consulte Parching for Express Brokers para obtener más información.

Información adicional sobre los corredores de Express

- Los corredores de Express trabajan con Apache Kafka APIs, pero aún no son totalmente compatibles con la KStreams API.
- Los corredores Express solo están disponibles en una AZs configuración de 3.
- Los corredores exprés solo están disponibles en tamaños de instancia seleccionados. Consulta los precios de Amazon MSK para ver la lista actualizada.
- Las versiones 3.6 y 3.8 de Apache Kafka admiten Express Brokers.

Tamaños del agente de Amazon MSK

Al crear un clúster aprovisionado de Amazon MSK, debe especificar el tamaño de los corredores que desea que tenga. Según el tipo de agente, Amazon MSK admite los siguientes tamaños de agente.

Tamaños de bróker estándar

- 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

Tamaños expresos de los corredores

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

> **Note**
>
> Es posible que algunos tamaños de corredores no estén disponibles en determinadas regiones. AWS Consulte las tablas de precios de instancias de bróker actualizadas en la página de precios de Amazon MSK para ver la lista más reciente de instancias disponibles por región.

Otras notas sobre el tamaño de los corredores

- Los corredores M7g utilizan procesadores AWS Graviton (procesadores personalizados basados en ARM creados por Amazon Web Services). Los agentes M7g ofrecen una mejor relación precio-rendimiento que las instancias M5 comparables. Los agentes M7g consumen menos energía que las instancias M5 comparables.
- Amazon MSK admite agentes M7g en clústeres aprovisionados de MSK que ejecuten las versiones 2.8.2 y 3.3.2 y superiores de Kafka.
- Los agentes M7g y M5 tienen un rendimiento de referencia más alto que los agentes T3 y se recomiendan para cargas de trabajo de producción. Los agentes M7g y M5 también pueden tener más particiones por agente que los agentes T3. Utilice los agentes M7g o M5 si está ejecutando cargas de trabajo de grado de producción más grandes o necesita un mayor número de particiones. Para obtener más información sobre los tamaños de las instancias M7g y M5, consulta Amazon EC2 General Purpose Instances.
- Los agentes T3 tienen la capacidad de usar créditos de CPU para rendimiento por ráfagas temporalmente. Utilice los agentes T3 para el desarrollo de bajo costo, si está probando cargas de trabajo de streaming de pequeñas a medianas, o si tiene cargas de trabajo de streaming de bajo rendimiento que experimentan picos temporales en el rendimiento. Le recomendamos que realice una proof-of-concept prueba para determinar si los intermediarios T3 son suficientes para la producción o para una carga de trabajo crítica. Para obtener más información sobre los tamaños de los corredores T3, consulte Amazon EC2 T3 Instances.

Para más información sobre cómo elegir los tamaños del agente, consulte Mejores prácticas para corredores de Standard y Express.

Gestión del almacenamiento para corredores estándar

Amazon MSK ofrece funciones que lo ayudan a administrar el almacenamiento en sus clústeres de MSK.

> **Note**
>
> Con Express Brokers, no necesita aprovisionar ni administrar ningún recurso de almacenamiento utilizado para sus datos. Esto simplifica la administración de los clústeres y elimina una de las causas más comunes de los problemas operativos de los clústeres de Apache Kafka. También gasta menos, ya que no tiene que aprovisionar capacidad de almacenamiento inactiva y solo paga por lo que usa.

Tipo de bróker estándar

Con los corredores estándar, puede elegir entre una variedad de opciones y capacidades de almacenamiento. Amazon MSK ofrece funciones que lo ayudan a administrar el almacenamiento en sus clústeres de MSK.

Para obtener información sobre la gestión del rendimiento, consulte???.

Temas

- Almacenamiento por niveles para corredores estándar
- Amplíe el almacenamiento de agentes estándar de Amazon MSK
- Gestione el rendimiento del almacenamiento para los agentes estándar en un clúster de Amazon MSK

Almacenamiento por niveles para corredores estándar

El almacenamiento por niveles es un nivel de almacenamiento de bajo costo para Amazon MSK que se puede escalar hasta ofrecer un almacenamiento prácticamente ilimitado, lo que permite crear aplicaciones de datos de streaming de manera rentable.

Puede crear un clúster de Amazon MSK y configurarlo con almacenamiento por niveles que equilibra el rendimiento y el costo. Amazon MSK almacena los datos de streaming en un nivel de almacenamiento principal optimizado para el rendimiento hasta que alcanzan los límites de retención

por temas de Apache Kafka. A continuación, Amazon MSK traslada automáticamente los datos al nuevo nivel de almacenamiento de bajo costo.

Cuando la aplicación empiece a leer los datos del almacenamiento por niveles, cabe esperar un aumento de la latencia de lectura durante los primeros bytes. A medida que empiece a leer los datos restantes de forma secuencial desde el nivel de bajo costo, cabe esperar latencias similares a las del nivel de almacenamiento principal. No es necesario aprovisionar almacenamiento para el almacenamiento por niveles de bajo costo ni administrar la infraestructura. Puede almacenar cualquier cantidad de datos y pagar únicamente por lo que utilice. Esta función es compatible con la APIs introducida en el KIP-405: Kafka Tiered Storage.

Para obtener información sobre el tamaño, la supervisión y la optimización de su clúster de almacenamiento en niveles de MSK, consulte Prácticas recomendadas para ejecutar cargas de trabajo de producción con el almacenamiento en niveles de Amazon MSK.

A continuación, se muestran algunas de las características del almacenamiento por niveles:

- Puede escalar a un almacenamiento prácticamente ilimitado, sin necesidad de adivinar cómo escalar la infraestructura de Apache Kafka.
- Puede retener los datos durante más tiempo en los temas de Apache Kafka o aumentar el almacenamiento de los temas sin necesidad de aumentar el número de agentes.
- Proporciona un búfer de seguridad de mayor duración para administrar los retrasos inesperados en el procesamiento.
- Puede volver a procesar los datos antiguos en su orden de producción exacto con el código de procesamiento de transmisiones existente y con Kafka. APIs
- Las particiones se vuelven a equilibrar más rápido porque no es necesario replicar los datos del almacenamiento secundario en los discos de los agentes.
- Los datos entre los agentes y el almacenamiento por niveles se trasladan en la VPC y no pasan por Internet.
- Un equipo cliente puede utilizar el mismo proceso para conectarse a clústeres nuevos con el almacenamiento por niveles habilitado que el que utiliza para conectarse a un clúster sin el almacenamiento por niveles habilitado. Consulte Crear un equipo cliente.

Requisitos de almacenamiento por niveles para los clústeres de Amazon MSK

- Debe utilizar la versión 3.0.0 o superior del cliente de Apache Kafka para crear un tema nuevo con el almacenamiento por niveles habilitado. Para hacer la transición de un tema existente a un

almacenamiento por niveles, puede volver a configurar un equipo cliente que utilice una versión de cliente de Kafka anterior a la 3.0.0 (la versión mínima admitida de Apache Kafka es la 2.8.2.tiered) para habilitar el almacenamiento por niveles. Consulte Paso 4: creación de un tema en el clúster de Amazon MSK.

- El clúster de Amazon MSK con el almacenamiento por niveles habilitado debe utilizar la versión 3.6.0 o superior, o la versión 2.8.2.tiered.

Restricciones y limitaciones de almacenamiento por niveles para clústeres de Amazon MSK

El almacenamiento por niveles tiene las siguientes restricciones y limitaciones:

- Asegúrese de que los clientes no estén configurados para `read_committed` al leer desde remote_tier en Amazon MSK, a menos que la aplicación utilice activamente la característica de transacciones.
- El almacenamiento por niveles no está disponible en las regiones (EE. UU. AWS GovCloud).
- El almacenamiento por niveles se aplica solo a los clústeres en modo aprovisionado.
- El almacenamiento por niveles no admite el tamaño del agente t3.small.
- El periodo mínimo de retención en el almacenamiento de bajo costo es de 3 días. No hay un periodo mínimo de retención para el almacenamiento principal.
- El almacenamiento por niveles no admite varios directorios de registros en un agente (características relacionadas con JBOD).
- El almacenamiento por niveles no admite temas compactados. Asegúrese de que todos los temas que tengan activado el almacenamiento por niveles tengan su cleanup.policy configurado únicamente para «ELIMINAR».
- El clúster de almacenamiento por niveles no permite modificar la política log.cleanup.policy de un tema una vez creado.
- El almacenamiento por niveles se puede deshabilitar para temas individuales, pero no para todo el clúster. Una vez deshabilitado, el almacenamiento por niveles no se puede volver a habilitar para un tema.
- Si utiliza la versión 2.8.2.tiered de Amazon MSK, solo podrá migrar sus datos a otra versión de Apache Kafka de almacenamiento por niveles compatible. Si no quiere seguir utilizando una versión de almacenamiento por niveles compatible, cree un nuevo clúster de MSK y migre sus datos a él.

- La kafka-log-dirs herramienta no puede informar sobre el tamaño de los datos de almacenamiento por niveles. La herramienta solo informa sobre el tamaño de los segmentos de registro en el almacenamiento principal.

Para obtener información sobre la configuración y las restricciones predeterminadas que debe tener en cuenta al configurar el almacenamiento por niveles a nivel temático, consulte. Directrices para la configuración a nivel de tema del almacenamiento en niveles de Amazon MSK

Cómo se copian los segmentos de registro al almacenamiento por niveles para un tema de Amazon MSK

Al habilitar el almacenamiento por niveles para un tema nuevo o existente, Apache Kafka copia los segmentos de registros cerrados del almacenamiento principal al almacenamiento por niveles.

- Apache Kafka solo copia los segmentos de registro cerrados. Copia todos los mensajes del segmento de registro en un almacenamiento por niveles.
- Los segmentos activos no son aptos para la organización por niveles. El tamaño de los segmentos de registro (segment.bytes) o el tiempo de traslado de los segmentos (segment.ms) controlan la velocidad de cierre de los segmentos y la velocidad a la que los copia Apache Kafka en el almacenamiento por niveles.

La configuración de retención de un tema con el almacenamiento por niveles habilitado es diferente de la configuración de un tema sin el almacenamiento por niveles habilitado. Las siguientes reglas controlan la retención de los mensajes en los temas con el almacenamiento por niveles habilitado:

- La retención se define en Apache Kafka con dos configuraciones: log.retention.ms (tiempo) y log.retention.bytes (tamaño). Estas configuraciones determinan la duración y el tamaño totales de los datos que retiene Apache Kafka en el clúster. Ya sea que habilite o no el modo de almacenamiento por niveles, estas configuraciones se establecen a nivel de clúster. Puede sustituir las configuraciones a nivel de tema con las configuraciones de tema.
- Al habilitar el almacenamiento por niveles, también puede especificar durante cuánto tiempo el nivel de almacenamiento principal de alto rendimiento almacena los datos. Por ejemplo, si un tema tiene una configuración de retención general (log.retention.ms) de 7 días y una retención local (local.retention.ms) de 12 horas, el almacenamiento principal del clúster retiene los datos solo durante las primeras 12 horas. El nivel de almacenamiento de bajo costo retiene los datos durante 7 días.

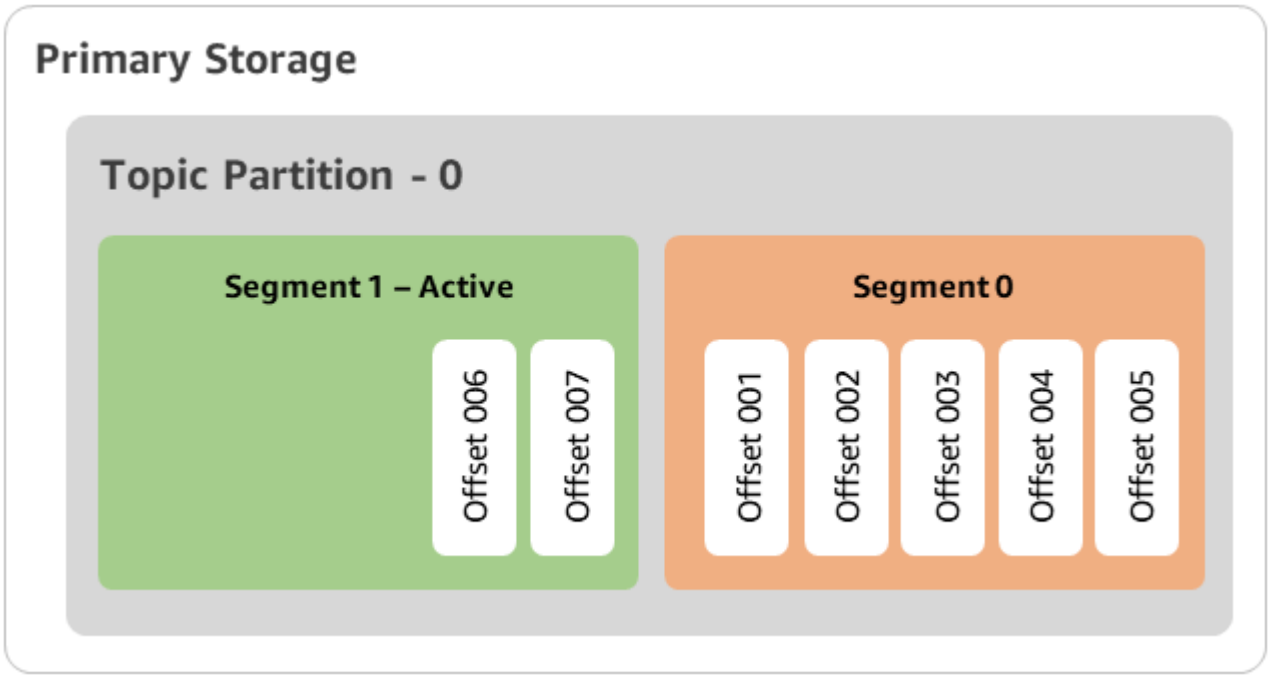
- Las configuraciones de retención habituales se aplican a todo el registro. Esto incluye sus partes por niveles y principales.
- La configuración local.retention.ms o local.retention.bytes controla la retención de los mensajes en el almacenamiento principal. Cuando los datos alcanzan los umbrales de la configuración de retención del almacenamiento principal (local.retention.ms/bytes) en un registro completo, Apache Kafka copia los datos del almacenamiento principal en un almacenamiento por niveles. En ese caso, los datos pueden caducar.
- Cuando Apache Kafka copia un mensaje de un segmento de registro en un almacenamiento por niveles, elimina el mensaje del clúster según la configuración retention.ms o retention.bytes.

Escenario de almacenamiento por niveles de Amazon MSK de ejemplo

Este escenario ilustra cómo se comporta un tema existente que tiene mensajes en el almacenamiento principal cuando el almacenamiento por niveles está habilitado. Para habilitar el almacenamiento por niveles en este tema, establezca remote.storage.enable en `true`. En este ejemplo, retention.ms se establece en 5 días y local.retention.ms se establece en 2 días. La siguiente es la secuencia de eventos cuando un segmento caduca.

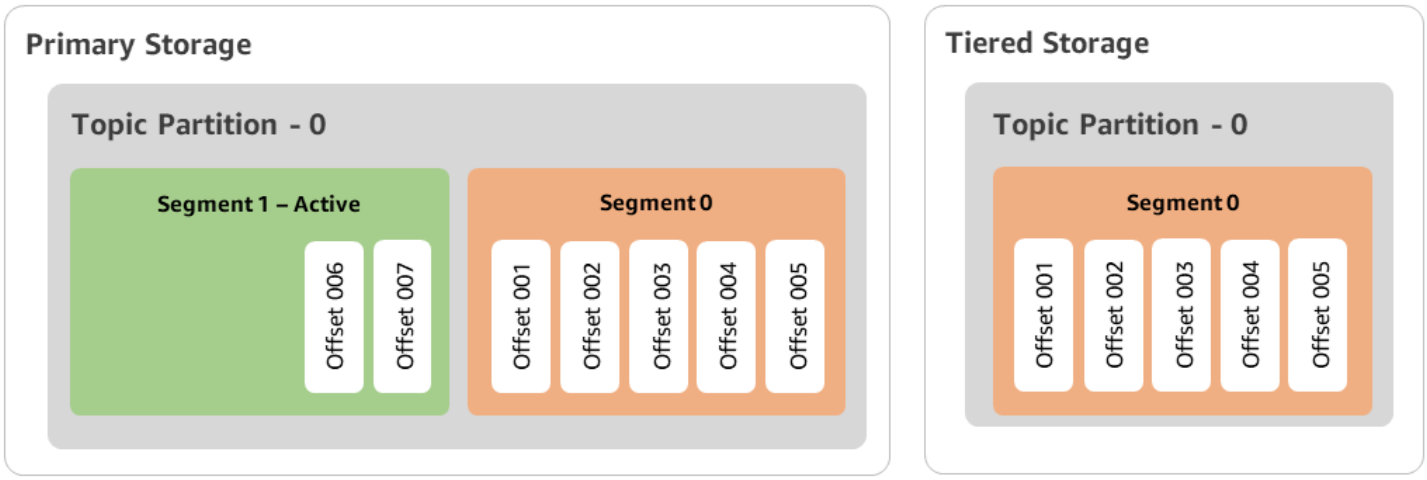
Momento T0: antes de habilitar el almacenamiento por niveles.

Antes de habilitar el almacenamiento por niveles para este tema, hay dos segmentos de registro. Uno de los segmentos está activo para la partición 0 de un tema existente.



Momento T1 (< 2 días): almacenamiento por niveles habilitado. El segmento 0 se copió en un almacenamiento por niveles.

Tras habilitar el almacenamiento por niveles para este tema, Apache Kafka copia el segmento 0 del registro en el almacenamiento por niveles una vez que el segmento cumple con la configuración de retención inicial. Apache Kafka también retiene la copia de almacenamiento principal del segmento 0. El segmento 1 activo aún no es apto para copiarse al almacenamiento por niveles. En este cronograma, Amazon MSK aún no aplica ninguna de las configuraciones de retención a ninguno de los mensajes del segmento 0 y del segmento 1. (retención local). bytes/ms, retention.ms/bytes)



Momento T2: retención local vigente.

Transcurridos 2 días, la configuración de retención principal entra en vigencia para el segmento 0 que copió Apache Kafka en el almacenamiento por niveles. Esto se determina si se establece local.retention.ms en 2 días. El segmento 0 ahora caduca a partir del almacenamiento principal. El segmento 1 activo aún no es apto para la caducidad ni se puede copiar en un almacenamiento por niveles.

Momento T3: retención general vigente.

Transcurridos 5 días, la configuración de retención entra en vigencia y Kafka borra el segmento 0 del registro y los mensajes asociados del almacenamiento por niveles. El segmento 1 aún no es apto para la caducidad ni se puede copiar en un almacenamiento por niveles porque está activo. El segmento 1 aún no está cerrado, por lo que no es apto para el traslado de segmentos.

Cree un clúster de Amazon MSK con almacenamiento por niveles con AWS Management Console

Este proceso describe cómo crear un clúster de Amazon MSK con almacenamiento por niveles mediante la AWS Management Console.

1. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/.
2. Elija Create cluster.
3. Seleccione Creación personalizada para el almacenamiento por niveles.

4. Especifique un nombre para el clúster.
5. En Tipo de clúster, seleccione Aprovisionado.
6. Elija una versión de Amazon Kafka que admita el almacenamiento por niveles para que Amazon MSK la utilice para crear el clúster.
7. Especifique un tamaño de agente distinto de kafka.t3.small.
8. Seleccione el número de agentes que quiere que Amazon MSK cree en cada zona de disponibilidad. El mínimo es un agente por zona de disponibilidad y el máximo es 30 agentes por clúster.
9. Indique el número de zonas en las que se distribuyen los agentes.
10. Indique el número de agentes de Apache Kafka que se implementan por zona.
11. Seleccione Opciones de almacenamiento. Esto incluye el almacenamiento por niveles y el almacenamiento de EBS para habilitar el modo de almacenamiento por niveles.
12. Siga el resto de pasos del asistente de creación de clúster. Al finalizar, el almacenamiento por niveles y el almacenamiento de EBS aparecen como el modo de almacenamiento del clúster en la vista Revisar y crear.
13. Seleccione Create cluster (Crear clúster).

Cree un clúster de Amazon MSK con almacenamiento por niveles con AWS CLI

Para habilitar el almacenamiento por niveles en un clúster, cree el clúster con la versión y el atributo correctos de Apache Kafka para el almacenamiento por niveles. Siga el ejemplo de código siguiente. Además, complete los pasos de la siguiente sección Cree un tema de Kafka con el almacenamiento por niveles habilitado con AWS CLI.

Consulte create-cluster para obtener una lista completa de los atributos admitidos para la creación de clústeres.

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

Cree un tema de Kafka con el almacenamiento por niveles habilitado con AWS CLI

Para finalizar el proceso que inició al crear un clúster con el almacenamiento por niveles habilitado, cree también un tema con el almacenamiento por niveles habilitado con los atributos del siguiente ejemplo de código. Los atributos específicos del almacenamiento por niveles son los siguientes:

- `local.retention.ms` (por ejemplo, 10 minutos) para la configuración de retención en función del tiempo o `local.retention.bytes` para los límites de tamaño de los segmentos de registro.
- `remote.storage.enable` se establece en `true` para habilitar el almacenamiento por niveles.

La siguiente configuración utiliza local.retention.ms, pero puede reemplazar este atributo por local.retention.bytes. Este atributo controla el tiempo que puede transcurrir o los bytes que Apache Kafka puede copiar antes de que Apache Kafka copie los datos del almacenamiento principal al almacenamiento por niveles. Consulte Topic-level configuration para más información sobre los atributos de configuración admitidos.

> **Note**
>
> Debe utilizar la versión 3.0.0 del cliente de Apache Kafka y superior. Estas versiones admiten una configuración denominada `remote.storage.enable` solo en esas versiones de cliente de `kafka-topics.sh`. Para habilitar el almacenamiento por niveles en un tema existente que utiliza una versión anterior de Apache Kafka, consulte la sección Habilitación del almacenamiento por niveles en un tema de Amazon MSK existente.

```
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
```

Habilitación y deshabilitación del almacenamiento por niveles en un tema de Amazon MSK existente

En estas secciones, se explica cómo habilitar y deshabilitar el almacenamiento por niveles en un tema que ya creó. Para crear un clúster y un tema nuevos con el almacenamiento por niveles habilitado, consulte Creating a cluster with tiered storage using the AWS Management Console.

Habilitación del almacenamiento por niveles en un tema de Amazon MSK existente

Para habilitar el almacenamiento por niveles en un tema existente, utilice la sintaxis de comando `alter` del ejemplo siguiente. Al habilitar el almacenamiento por niveles en un tema ya existente, no tendrá que limitarse a una versión determinada del cliente de Apache Kafka.

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

Deshabilitación del almacenamiento por niveles en un tema de Amazon MSK existente

Para deshabilitar el almacenamiento por niveles en un tema existente, utilice la sintaxis del comando `alter` en el mismo orden en que lo utilizó cuando habilitó el almacenamiento por niveles.

```
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**
>
> Al deshabilitar el almacenamiento por niveles, se eliminan por completo los datos del tema del almacenamiento por niveles. Apache Kafka retiene los datos del almacenamiento principal, pero sigue aplicando las reglas de retención principal en función de `local.retention.ms`. Una vez deshabilitado el almacenamiento por niveles en un tema, no se puede volver a habilitar. Si quiere deshabilitar el almacenamiento por niveles en un tema existente, no tendrá que limitarse a una versión determinada del cliente de Apache Kafka.

Habilite el almacenamiento por niveles en un clúster de Amazon MSK existente mediante CLI AWS

> **Note**
>
> Puede habilitar el almacenamiento por niveles solo si log.cleanup.policy del clúster está establecida en `delete`, ya que el almacenamiento por niveles no admite temas compactados. Más adelante, podrá configurar log.cleanup.policy de un tema individual en `compact` si el almacenamiento por niveles no está habilitado en ese tema concreto.

Consulte Topic-level configuration para más información sobre los atributos de configuración admitidos.

1. Actualice la versión de Kafka: las versiones en clúster no son números enteros simples. Para buscar la versión actual del clúster, utilice la `DescribeCluster` operación o el comando `describe-cluster` AWS CLI. Un ejemplo de ID de versión es `KTVPDKIKX0DER`.

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

2. Edite el modo de almacenamiento del clúster. El siguiente ejemplo de código muestra cómo cambiar el modo de almacenamiento del clúster a `TIERED` mediante la API `update-storage`.

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

Actualización del almacenamiento por niveles en un clúster de Amazon MSK existente mediante la consola

Este proceso describe cómo actualizar un clúster de Amazon MSK de almacenamiento por niveles mediante la AWS Management Console.

Asegúrese de que la versión actual de Apache Kafka del clúster de MSK sea 2.8.2.tiered. Consulte la página sobre la actualización de la versión de Apache Kafka si debe actualizar el clúster de MSK a la versión 2.8.2.tiered.

> **Note**
>
> Puede habilitar el almacenamiento por niveles solo si log.cleanup.policy del clúster está establecida en `delete`, ya que el almacenamiento por niveles no admite temas compactados. Más adelante, podrá configurar log.cleanup.policy de un tema individual en `compact` si el almacenamiento por niveles no está habilitado en ese tema concreto. Consulte Topic-level configuration para más información sobre los atributos de configuración admitidos.

1. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/.

2. Vaya a la página de resumen del clúster y elija Propiedades.
3. Vaya a la sección Almacenamiento y elija Editar modo de almacenamiento de clúster.
4. Elija Almacenamiento por niveles y almacenamiento de EBS y, luego, Guardar cambios.

Amplíe el almacenamiento de agentes estándar de Amazon MSK

Puede ampliar la cantidad de almacenamiento de EBS por agente. No puede disminuir el almacenamiento.

Los volúmenes de almacenamiento siguen estando disponibles durante esta operación de ampliación.

> **Important**
>
> Cuando el almacenamiento se escala para un clúster de MSK, el almacenamiento adicional está disponible de inmediato. Sin embargo, el clúster requiere un periodo de recuperación después de cada evento de escalado del almacenamiento. Amazon MSK utiliza este periodo de recuperación para optimizar el clúster y poder escalarlo de nuevo. Este periodo puede oscilar entre un mínimo de 6 horas y más de 24 horas, en función del tamaño y la utilización del almacenamiento del clúster y en función del tráfico. Esto es aplicable tanto a los eventos de escalado automático como al escalado manual mediante la UpdateBrokerStorageoperación. Para obtener información sobre el tamaño correcto del almacenamiento, consulte the section called “Prácticas recomendadas para agentes”.

Puede utilizar el almacenamiento por niveles para escalar verticalmente a cantidades ilimitadas de almacenamiento para el agente. Consulte, Almacenamiento por niveles para corredores estándar.

Temas

- Escalado automático para clústeres de Amazon MSK
- Escalado manual para corredores estándar

Escalado automático para clústeres de Amazon MSK

Para ampliar automáticamente el almacenamiento del clúster en respuesta al aumento del uso, puede configurar una política de escalado automático de aplicaciones para Amazon MSK. En una

política de escalado automático, se establece la utilización del disco objetivo y la capacidad máxima de escalado.

Antes de utilizar el escalado automático para Amazon MSK, debe tener en cuenta lo siguiente:

- > **Important**
  >
  > Una acción de escalado del almacenamiento solo puede producirse una vez cada seis horas.

  Le recomendamos que comience con un volumen de almacenamiento del tamaño adecuado para sus demandas de almacenamiento. Para obtener orientación sobre el tamaño correcto de su clúster, consulte Ajuste el tamaño correcto de su clúster: número de agentes estándar por clúster.
- Amazon MSK no reduce el almacenamiento de clústeres en respuesta a la reducción del uso. Amazon MSK no admite reducir el tamaño de los volúmenes de almacenamiento. Si necesita reducir el tamaño del almacenamiento en clúster, debe migrar el clúster existente a un clúster con un almacenamiento más pequeño. Para obtener más información sobre cómo migrar un clúster, consulte Migración a un clúster de Amazon MSK.
- Amazon MSK no admite el escalado automático en las regiones Asia-Pacífico (Osaka) y África (Ciudad del Cabo).
- Al asociar una política de autoescalado a su clúster, Amazon EC2 Auto Scaling crea automáticamente una CloudWatch alarma de Amazon para el seguimiento de los objetivos. Si eliminas un clúster con una política de autoescalado, esta CloudWatch alarma persiste. Para eliminar la CloudWatch alarma, debe eliminar una política de autoescalado de un clúster antes de eliminarlo. Para obtener más información sobre el seguimiento de objetivos, consulte Políticas de escalado del seguimiento de objetivos para Amazon EC2 Auto Scaling en la Guía del usuario de Amazon EC2 Auto Scaling.

Temas

- Detalles de la política de escalado automático de Amazon MSK
- Configuración del escalado automático para su clúster de Amazon MSK

Detalles de la política de escalado automático de Amazon MSK

La política de escalado automático define la siguiente métrica predefinida para el clúster:

- Objetivo de utilización de almacenamiento: el umbral de utilización de almacenamiento que Amazon MSK utiliza para activar una operación de escalado automático. Puede establecer el objetivo de utilización entre el 10 y el 80 % de la capacidad de almacenamiento actual. Le recomendamos que habilite el objetivo de utilización del almacenamiento entre el 50 % y el 60 %.
- Capacidad máxima de almacenamiento: el límite máximo de escalado que Amazon MSK puede establecer para el almacenamiento de su agente. Puede configurar una capacidad máxima de almacenamiento de hasta 16 TiB por agente. Para obtener más información, consulte Cuota de Amazon MSK.

Cuando Amazon MSK detecta que la métrica `Maximum Disk Utilization` es igual o superior a la configuración `Storage Utilization Target`, aumenta la capacidad de almacenamiento en una cantidad igual al mayor de los dos números: 10 GiB o el 10 % del almacenamiento actual. Por ejemplo, si tiene 1000 GiB, esa cantidad es de 100 GiB. El servicio comprueba el uso del almacenamiento cada minuto. Las operaciones de escalado adicionales siguen aumentando el almacenamiento en una cantidad igual al mayor de los dos números: 10 GiB o el 10 % del almacenamiento actual.

Para determinar si se han realizado operaciones de autoescalado, utilice la ListClusterOperationsoperación.

Configuración del escalado automático para su clúster de Amazon MSK

Puede utilizar la consola de Amazon MSK, la API de Amazon MSK o AWS CloudFormation implementar el escalado automático del almacenamiento. CloudFormation el soporte está disponible a través de. Application Auto Scaling

> **Note**
>
> No puede implementar el escalado automático cuando crea un clúster. Primero debe crear el clúster y, a continuación, crear y habilitar una política de escalado automático para él. Sin embargo, puede crear la política mientras el servicio Amazon MSK crea el clúster.

Temas

- Configuración del escalado automático desde la AWS Management Console de Amazon MSK
- Configure el escalado automático mediante la CLI
- Configuración del escalado automático de Amazon MSK con la API

Configuración del escalado automático desde la AWS Management Console de Amazon MSK

En este proceso, se describe cómo utilizar la consola de Amazon MSK para implementar el escalado automático para almacenamiento.

1. ¿Iniciar sesión en la AWS Management Console consola Amazon MSK y abrirla desde https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. En la lista de clústeres, elija su clúster. Se le redigirá a una página en la que se enumeran los detalles del clúster.
3. En la sección Escalado automático para el almacenamiento, elija Configurar.
4. Cree una política de escalado automático y asígnele un nombre. Especifique el objetivo de utilización del almacenamiento, la capacidad máxima de almacenamiento y la métrica objetivo.
5. Elija `Save changes`.

Al guardar y habilitar la nueva política, la política se activa para el clúster. A continuación, Amazon MSK amplía el almacenamiento del clúster cuando se alcanza el objetivo de utilización del almacenamiento.

Configure el escalado automático mediante la CLI

En este proceso, se describe cómo utilizar la CLI de Amazon MSK para implementar el escalado automático para almacenamiento.

1. Utilice el `RegisterScalableTarget`comando para registrar un objetivo de utilización del almacenamiento.
2. Utilice el `PutScalingPolicy`comando para crear una política de expansión automática.

Configuración del escalado automático de Amazon MSK con la API

En este proceso, se describe cómo implementar el escalado automático del almacenamiento con la API de Amazon MSK.

1. Utilice la `RegisterScalableTarget`API para registrar un objetivo de utilización del almacenamiento.
2. Usa la `PutScalingPolicy`API para crear una política de expansión automática.

Escalado manual para corredores estándar

Para aumentar el almacenamiento, espere a que el clúster esté en el estado `ACTIVE`. El escalado del almacenamiento tiene un periodo de recuperación de al menos seis horas entre eventos. Aunque la operación permite disponer de almacenamiento adicional de forma inmediata, el servicio realiza optimizaciones en el clúster que pueden tardar hasta 24 horas o más. La duración de estas optimizaciones es proporcional al tamaño del almacenamiento.

Ampliar el almacenamiento de los corredores mediante el AWS Management Console

1. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/.
2. Elija el clúster de MSK para actualizar el almacenamiento del agente.
3. En la sección Almacenamiento elija Editar.
4. Especifique el volumen de almacenamiento que desee. Solo se puede aumentar la cantidad de almacenamiento, no se puede disminuir.
5. Seleccione Save changes (Guardar cambios).

Ampliar el almacenamiento de agentes mediante el AWS CLI

Ejecute el siguiente comando, reemplazando *`ClusterArn`*por el nombre de recurso de Amazon (ARN) que obtuvo al crear el clúster. Si no tiene el ARN para su clúster, puede encontrarlo enumerando todos los clústeres. Para obtener más información, consulte the section called “Enumerar clústeres”.

Reemplace *`Current-Cluster-Version`* con la versión actual del clúster.

> **Important**
>
> Las versiones de clúster no son enteros simples. Para encontrar la versión actual del clúster, utilice la DescribeClusteroperación o el comando AWS CLI describe-cluster. Un ejemplo de ID de versión es `KTVPDKIKX0DER`.

El *`Target-Volume-in-GiB`* parámetro representa la cantidad de almacenamiento que desea que tenga cada agente. Sólo es posible actualizar el almacenamiento para todos los agentes. No puede especificar agentes individuales para los que actualizar el almacenamiento. El valor que especifique *`Target-Volume-in-GiB`* debe ser un número entero superior a 100 GiB. El almacenamiento por agente después de la operación de actualización no puede exceder los 16384 GiB.

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

Escalar verticalmente el almacenamiento del agente mediante la API

Para actualizar el almacenamiento de un broker mediante la API, consulte UpdateBrokerStorage.

Gestione el rendimiento del almacenamiento para los agentes estándar en un clúster de Amazon MSK

Para obtener información sobre cómo aprovisionar el rendimiento mediante la consola, la CLI y la API de Amazon MSK, consulte. ???

Temas

- Cuellos de botella en el rendimiento del agente de Amazon MSK y configuración de rendimiento máximo
- Medición del rendimiento del almacenamiento de un clúster de Amazon MSK
- Valores de actualización de configuración para el almacenamiento aprovisionado en un clúster de Amazon MSK
- Aprovisione el rendimiento del almacenamiento para los corredores estándar en un clúster de Amazon MSK

Cuellos de botella en el rendimiento del agente de Amazon MSK y configuración de rendimiento máximo

Hay varias causas de los cuellos de botella en el rendimiento de los corredores: el rendimiento del volumen, el rendimiento de la red de Amazon a EC2 Amazon EBS y el rendimiento de salida de Amazon. EC2 Puede habilitar el rendimiento del almacenamiento aprovisionado para ajustar el rendimiento del volumen. Sin embargo, las limitaciones de rendimiento de los corredores pueden deberse al rendimiento de la red de Amazon EC2 a Amazon EBS y al rendimiento de salida de Amazon EC2.

El rendimiento de EC2 salida de Amazon se ve afectado por la cantidad de grupos de consumidores y de consumidores por grupo de consumidores. Además, tanto el rendimiento de la red Amazon EC2 a Amazon EBS como el rendimiento de EC2 salida de Amazon son más altos para los corredores de mayor tamaño.

Para volúmenes de 10 GiB o más, puede aprovisionar un rendimiento de almacenamiento de 250 MiB por segundo o más. 250 MiB por segundo es el valor predeterminado. Para aprovisionar el rendimiento del almacenamiento, debe elegir el tamaño del agente kafka.m5.4xlarge o superior (o kafka.m7g.2xlarge o superior) y puede especificar el rendimiento máximo, como se muestra en la siguiente tabla.

tamaño del agente	Rendimiento máximo de almacenamiento (MiB/segundo)
kafka.m5.4xlarge	593
kafka.m5.8xlarge	850
kafka.m5.12xlarge	1 000
kafka.m5.16xlarge	1 000
kafka.m5.24xlarge	1 000
kafka.m7g.2xlarge	312,5
kafka.m7g.4xlarge	625
kafka.m7g.8xlarge	1 000
kafka.m7g.12xlarge	1 000
kafka.m7g.16xlarge	1 000

Medición del rendimiento del almacenamiento de un clúster de Amazon MSK

Puede utilizar las métricas `VolumeReadBytes` y `VolumeWriteBytes` para medir el rendimiento medio de almacenamiento de un clúster. La suma de estas dos métricas proporciona el rendimiento de almacenamiento medio en bytes. Para obtener el rendimiento de almacenamiento medio de un clúster, establezca estas dos métricas en SUM y el periodo en 1 minuto y, luego, utilice la siguiente fórmula.

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

Para obtener información sobre las métricas `VolumeReadBytes` y `VolumeWriteBytes`, consulte the section called “Supervisión de PER_BROKER”.

Valores de actualización de configuración para el almacenamiento aprovisionado en un clúster de Amazon MSK

Puede actualizar la configuración de Amazon MSK antes o después de activar el rendimiento aprovisionado. Sin embargo, no verá el rendimiento deseado hasta que realice ambas acciones: actualice el parámetro de configuración `num.replica.fetchers` y active el rendimiento aprovisionado.

En la configuración predeterminada de Amazon MSK, `num.replica.fetchers` tiene un valor de 2. Para actualizar `num.replica.fetchers`, puede utilizar los valores sugeridos de la siguiente tabla. Estos valores son orientativos. Le recomendamos que ajuste estos valores en función del caso de uso.

tamaño del agente	num.replica.fetchers
kafka.m5.4xlarge	4
kafka.m5.8xlarge	8
kafka.m5.12xlarge	14
kafka.m5.16xlarge	16
kafka.m5.24xlarge	16

Es posible que la configuración actualizada no surta efecto hasta dentro de 24 horas y que tarde más si el volumen de origen no se utiliza por completo. Sin embargo, el rendimiento del volumen transitorio es como mínimo igual al rendimiento de los volúmenes de almacenamiento de origen durante el periodo de migración. Un volumen de 1 TiB totalmente utilizado tarda normalmente unas seis horas en migrar a una configuración actualizada.

Aprovisione el rendimiento del almacenamiento para los corredores estándar en un clúster de Amazon MSK

Los agentes de Amazon MSK mantienen los datos de los volúmenes de almacenamiento. I/O El almacenamiento se consume cuando los productores escriben en el clúster, cuando los datos se

replican entre intermediarios y cuando los consumidores leen datos que no están en la memoria. El rendimiento del almacenamiento de volúmenes es la velocidad a la que se pueden escribir datos en un volumen de almacenamiento, y leerse desde este. El rendimiento del almacenamiento aprovisionado es la capacidad de especificar esa velocidad para los agentes del clúster.

Puede especificar la velocidad de rendimiento aprovisionado en MiB por segundo para los clústeres cuyos agentes sean de tamaño `kafka.m5.4xlarge` o más grandes y si el volumen de almacenamiento es de 10 GiB o superior. Es posible especificar el rendimiento aprovisionado durante la creación del clúster. También puede habilitar o deshabilitar el rendimiento aprovisionado para un clúster que tenga el estado `ACTIVE`.

Para obtener información sobre la administración del rendimiento, consulte. ???

Temas

- Aprovisione el rendimiento del almacenamiento del clúster de Amazon MSK mediante AWS Management Console
- Aprovisione el rendimiento del almacenamiento del clúster de Amazon MSK mediante AWS CLI
- Aprovisionamiento de rendimiento de almacenamiento al crear un clúster de Amazon MSK mediante la API

Aprovisione el rendimiento del almacenamiento del clúster de Amazon MSK mediante AWS Management Console

Este proceso muestra un ejemplo de cómo puede usarlo AWS Management Console para crear un clúster de Amazon MSK con el rendimiento aprovisionado habilitado.

1. ¿Iniciar sesión en la AWS Management Console consola Amazon MSK y abrirla desde https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. Elija Create cluster.
3. Seleccione Creación personalizada.
4. Especifique un nombre para el clúster.
5. En la sección Almacenamiento elija Habilitar.
6. Elija un valor en Rendimiento de almacenamiento por agente.
7. Elija una VPC, zonas y subredes y un grupo de seguridad.
8. Elija Siguiente.
9. Al final del paso Seguridad, elija Siguiente.

10. En final del paso Supervisión y etiquetas, elija Siguiente.
11. Revise la configuración del clúster y seleccione Crear.

Aprovisione el rendimiento del almacenamiento del clúster de Amazon MSK mediante AWS CLI

Este proceso muestra un ejemplo de cómo se puede utilizar AWS CLI para crear un clúster con el rendimiento aprovisionado activado.

1. Copie el siguiente JSON y péguelo en un archivo. Sustituya los marcadores de posición de los identificadores de subred IDs y grupo de seguridad por los valores de su cuenta. Asigne el nombre `cluster-creation.json` al archivo y guárdelo.

```
{
    "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. Ejecute el siguiente AWS CLI comando desde el directorio en el que guardó el archivo JSON en el paso anterior.

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

Aprovisionamiento de rendimiento de almacenamiento al crear un clúster de Amazon MSK mediante la API

Para configurar el rendimiento del almacenamiento aprovisionado al crear un clúster, utilice la V2. CreateCluster

Seguridad en Amazon MSK

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de una arquitectura de centro de datos y red diseñada para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre usted AWS y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWS programas de de . Para obtener información sobre los programas de conformidad que se aplican a Amazon Managed Streaming para Apache Kafka, consulte Servicios de Amazon Web Services en el ámbito del programa de conformidad.
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos vigentes.

Esta documentación le permite comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza Amazon MSK. En los siguientes temas, se mostrará cómo configurar Amazon MSK para satisfacer sus objetivos de seguridad y conformidad. También puede aprender a utilizar otros servicios de Amazon Web Services que ayudan a supervisar y proteger los recursos de Amazon MSK.

Temas

- Protección de datos en Amazon Managed Streaming para Apache Kafka
- Autenticación y autorización para Amazon MSK APIs
- Autenticación y autorización para Apache Kafka APIs
- Modificación del grupo de seguridad de un clúster de Amazon MSK
- Controle el acceso a ZooKeeper los nodos de Apache en su clúster de Amazon MSK
- Validación de la conformidad de Amazon Managed Streaming para Apache Kafka
- Resiliencia en Amazon Managed Streaming para Apache Kafka
- Seguridad de la infraestructura en Amazon Managed Streaming para Apache Kafka

Protección de datos en Amazon Managed Streaming para Apache Kafka

El modelo de se aplica a protección de datos en Amazon Managed Streaming for Apache Kafka. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los Nube de AWS. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulta las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulta la publicación de blog sobre el Modelo de responsabilidad compartida de AWS y GDPR en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Se utiliza SSL/TLS para comunicarse con AWS los recursos. Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte Cómo trabajar con CloudTrail senderos en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.

- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-3 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulta Estándar de procesamiento de la información federal (FIPS) 140-3.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con Amazon MSK u otro Servicios de AWS mediante la consola, la API o AWS SDKs. AWS CLI Cualquier dato que ingrese en etiquetas o campos de texto de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Temas

- Cifrado de Amazon MSK
- Introducción al cifrado de Amazon MSK
- Utilice Amazon MSK APIs con puntos de enlace de VPC de interfaz

Cifrado de Amazon MSK

Amazon MSK proporciona opciones de cifrado de datos que puede utilizar para cumplir estrictos requisitos de administración de datos. Los certificados que Amazon MSK utiliza para el cifrado deben renovarse cada 13 meses. Amazon MSK renueva automáticamente estos certificados para todos los clústeres. Los clústeres de corredores de Express permanecen en `ACTIVE` estado cuando Amazon MSK inicia la operación de actualización de certificados. En el caso de los clústeres de corredores estándar, Amazon MSK establece el estado del clúster en el que `MAINTENANCE` se inicia la operación de actualización de certificados. MSK lo restablece en el momento en que `ACTIVE` finaliza la actualización. Mientras un clúster se encuentre en la operación de actualización de certificados, puede seguir produciendo y consumiendo datos, pero no podrá realizar ninguna operación de actualización en ellos.

Cifrado en reposo de Amazon MSK

Amazon MSK se integra con AWS Key Management Service (KMS) para ofrecer cifrado transparente del servidor. Amazon MSK siempre cifra sus datos en reposo. Al crear un clúster de MSK, puede especificar la propiedad AWS KMS key que desea que Amazon MSK utilice para cifrar sus datos en reposo. Si no se especifica una clave de KMS, Amazon MSK crea una administrada por Clave administrada de AWS y la utiliza en su nombre. Para obtener más información acerca de las claves de KMS, consulte AWS KMS keys en la Guía para desarrolladores de AWS Key Management Service .

Cifrado en tránsito de Amazon MSK

Amazon MSK utiliza TLS 1.2. De forma predeterminada, cifra los datos en tránsito entre los agentes de su clúster de MSK. Puede anular este valor predeterminado en el momento en que cree el clúster.

Para la comunicación entre clientes y agentes, debe especificar una de las tres opciones siguientes:

- Permitir solo datos cifrados TLS. Este es el valor predeterminado.
- Permitir tanto datos de texto sin formato como datos cifrados TLS.
- Permitir solo datos de texto sin formato.

Los corredores de Amazon MSK utilizan AWS Certificate Manager certificados públicos. Por lo tanto, cualquier almacén de confianza que confíe en Amazon Trust Services también confía en los certificados de los agentes de Amazon MSK.

Si bien recomendamos encarecidamente habilitar el cifrado en tránsito, puede agregar sobrecarga de CPU adicional y unos pocos milisegundos de latencia. Sin embargo, la mayoría de los casos de uso no son sensibles a estas diferencias y la magnitud del impacto depende de la configuración del clúster, los clientes y el perfil de uso.

Introducción al cifrado de Amazon MSK

Al crear un clúster de MSK, puede especificar la configuración de cifrado en formato JSON. A continuación se muestra un ejemplo.

```
{
   "EncryptionAtRest": {
       "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/abcdabcd-1234-
abcd-1234-abcd123e8e8e"
    },
```

```
    "EncryptionInTransit": {
         "InCluster": true,
         "ClientBroker": "TLS"
     }
}
```

Para `DataVolumeKMSKeyId`, puede especificar una clave administrada por el cliente o la Clave administrada de AWS para MSK en su cuenta (`alias/aws/kafka`). Si no lo especifica`EncryptionAtRest`, Amazon MSK seguirá cifrando sus datos en reposo en. Clave administrada de AWS Para determinar qué clave está utilizando su clúster, envíe una solicitud `GET` o invoque la operación de la API de `DescribeCluster`.

En `EncryptionInTransit`, el valor predeterminado de `InCluster` es true, pero puede establecerlo en false si no desea que Amazon MSK cifre sus datos a medida que pasan entre los agentes.

Para especificar el modo de cifrado de los datos en tránsito entre clientes y agentes, establezca `ClientBroker` a uno de los tres valores: `TLS`, `TLS_PLAINTEXT`, o `PLAINTEXT`.

Temas

- Especificación de la configuración de cifrado al crear un clúster de Amazon MSK
- Prueba del cifrado TLS de Amazon MSK

Especificación de la configuración de cifrado al crear un clúster de Amazon MSK

En este proceso, se describe cómo especificar una configuración de cifrado al crear un clúster de Amazon MSK.

Especificación de la configuración de cifrado al crear un clúster

1. Guarde el contenido del ejemplo anterior en un archivo y asígnele el nombre que desee. Por ejemplo, llámalo `encryption-settings.json`.
2. Ejecute el comando `create-cluster` y use la opción `encryption-info` para señalar al archivo donde guardó su configuración JSON. A continuación se muestra un ejemplo. Sustitúyala por una versión que coincida *`{YOUR MSK VERSION}`* con la versión del cliente de Apache Kafka. Para obtener información sobre cómo encontrar la versión de clúster de MSK, consulte Determinar la versión del clúster de MSK. Tenga en cuenta que el uso de una versión de cliente de Apache Kafka que no sea la misma que su versión de clúster de MSK puede provocar la corrupción, la pérdida y el tiempo de inactividad de los datos de Apache Kafka.

```
aws kafka create-cluster --cluster-name "ExampleClusterName" --broker-node-group-
info file://brokernodegroupinfo.json --encryption-info file://encryptioninfo.json
 --kafka-version "{YOUR MSK VERSION}" --number-of-broker-nodes 3
```

El siguiente es un ejemplo de una respuesta correcta después de ejecutar este comando.

```
{
    "ClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/SecondTLSTest/
abcdabcd-1234-abcd-1234-abcd123e8e8e",
    "ClusterName": "ExampleClusterName",
    "State": "CREATING"
}
```

Prueba del cifrado TLS de Amazon MSK

En este proceso, se describe cómo probar el cifrado TLS con Amazon MSK.

Prueba del cifrado TLS

1. Cree un equipo cliente siguiendo las instrucciones de the section called “Creación de un equipo cliente”.
2. Instale Apache Kafka en el equipo cliente.
3. En este ejemplo, usamos el almacén de confianza de JVM para comunicarnos con el clúster de MSK. Para ello, primero cree una carpeta denominada `/tmp` en el equipo cliente. Luego, vaya a la carpeta `bin` de la instalación de Apache Kafka y ejecute el siguiente comando. (Su ruta de JVM puede ser diferente).

   ```
   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. Mientras esté aún en la carpeta `bin` de la instalación de Apache Kafka en el equipo cliente, cree un archivo de texto denominado `client.properties` con el siguiente contenido.

   ```
   security.protocol=SSL
   ssl.truststore.location=/tmp/kafka.client.truststore.jks
   ```

5. Ejecute el siguiente comando en una máquina que lo tenga AWS CLI instalado y *clusterARN* reemplácelo por el ARN de su clúster.

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

Un resultado correcto sería como el siguiente. Guarde este resultado porque lo necesita para el siguiente paso.

```
{
    "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. Ejecute el siguiente comando y *BootstrapBrokerStringTls* reemplácelo por uno de los puntos finales del broker que obtuvo en el paso anterior.

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

7. Abra una nueva ventana de comandos y conéctese al mismo equipo cliente. A continuación, ejecute el siguiente comando para crear un consumidor de consola.

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBrokerStringTls --consumer.config client.properties --topic
 TLSTestTopic
```

8. En la ventana del productor, escriba un mensaje de texto seguido de una devolución y busque el mismo mensaje en la ventana del consumidor. Amazon MSK cifró este mensaje en tránsito.

Para obtener más información acerca de cómo configurar clientes Apache Kafka para que funcionen con datos cifrados, consulte Configuración de clientes Kafka.

Utilice Amazon MSK APIs con puntos de enlace de VPC de interfaz

Puede utilizar un punto de enlace de interfaz VPC, con tecnología de AWS PrivateLink, para evitar que el tráfico entre su Amazon VPC y Amazon APIs MSK salga de la red de Amazon. Los puntos finales de interfaz VPC no requieren una puerta de enlace a Internet, un dispositivo NAT, una conexión VPN o una conexión Direct AWS Connect. AWS PrivateLinkes una AWS tecnología que permite la comunicación privada entre AWS servicios mediante una interfaz de red elástica y privada

IPs en su Amazon VPC. Para obtener más información, consulte Amazon Virtual Private Cloud e Interface VPC Endpoints ().AWS PrivateLink

Sus aplicaciones se pueden conectar con Amazon MSK Provisioned y MSK Connect APIs mediante. AWS PrivateLink Para empezar, cree un punto de enlace de VPC de interfaz para su API de Amazon MSK a fin de iniciar el flujo de tráfico desde y hacia sus recursos de Amazon VPC a través del punto de enlace de VPC de interfaz. Los puntos finales de VPC con interfaz habilitada para FIPS están disponibles para las regiones de EE. UU. Para obtener más información, consulte Crear un punto final de interfaz.

Con esta función, sus clientes de Apache Kafka pueden obtener dinámicamente las cadenas de conexión para conectarse con los recursos de MSK Provisioned o MSK Connect sin tener que atravesar Internet para recuperar las cadenas de conexión.

Al crear un punto final de VPC de interfaz, elija uno de los siguientes puntos de enlace de nombre de servicio:

Para MSK Provisioned:

- com.amazonaws.region.kafka
- com.amazonaws.region.kafka-fips (compatible con FIPS)

Dónde región es el nombre de tu región. Elija este nombre de servicio para que funcione con MSK Provisioned compatible. APIs Para obtener más información, consulte Operaciones en la versión 1.0/apireference/. https://docs.aws.amazon.com/msk/

Para MSK Connect:

- com.amazonaws.region.kafkaconnect

Dónde región es el nombre de tu región. Elija este nombre de servicio para que funcione con MSK Connect APIs compatible. Para obtener más información, consulte Acciones en la referencia de la API de Amazon MSK Connect.

Para obtener más información, incluidas step-by-step las instrucciones para crear un punto final de VPC de interfaz, consulte Creación de un punto final de interfaz en la AWS PrivateLink Guía.

Controle el acceso a los puntos de enlace de VPC para Amazon MSK Provisioned o MSK Connect APIs

Las políticas de punto de conexión de VPC le permiten controlar el acceso asociando una política a un punto de conexión de VPC o utilizando campos adicionales en una política asociada a un usuario, grupo o rol de IAM para restringir el acceso para que solo se produzca a través del punto de conexión de VPC especificado. Use la política de ejemplo adecuada para definir los permisos de acceso para el servicio MSK Provisioned o MSK Connect.

Si no adjunta una política al crear un punto de conexión, Amazon VPC adjunta una política predeterminada que le conceda acceso completo al servicio. Una política de punto de conexión no anula ni reemplaza las políticas basadas en identidad de IAM ni las políticas específicas del servicio. Se trata de una política independiente para controlar el acceso desde el punto de conexión al servicio especificado.

Para obtener más información, consulte Control del acceso a los servicios con puntos de enlace de VPC en la guía.AWS PrivateLink

MSK Provisioned — VPC policy example

Acceso de solo lectura

Este ejemplo de política se puede adjuntar a un punto final de VPC. (Para obtener más información, consulte Control de acceso a recursos de Amazon VPC). Restringe las acciones a enumerar y describir únicamente las operaciones a través del punto final de la VPC al que está conectado.

```
{
  "Statement": [
    {
      "Sid": "MSKReadOnly",
      "Principal": "*",
      "Action": [
        "kafka:List*",
        "kafka:Describe*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

MSK Provisioned: ejemplo de política de punto final de VPC

Restrinja el acceso a un clúster de MSK específico

Este ejemplo de política se puede adjuntar a un punto final de VPC. Restringe el acceso a un clúster de Kafka específico a través del punto final de la VPC al que está conectado.

```
{
  "Statement": [
    {
      "Sid": "AccessToSpecificCluster",
      "Principal": "*",
      "Action": "kafka:*",
      "Effect": "Allow",
      "Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/MyCluster"
    }
  ]
}
```

MSK Connect — VPC endpoint policy example

Enumere los conectores y cree uno nuevo

El siguiente es un ejemplo de una política de punto final para MSK Connect. Esta política permite al rol especificado enumerar los conectores y crear un conector nuevo.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "MSKConnectPermissions",
            "Effect": "Allow",
            "Action": [
                "kafkaconnect:ListConnectors",
                "kafkaconnect:CreateConnector"
            ],
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:role/<ExampleRole>"
                ]
            }
        }
```

```
    ]
}
```

MSK Connect: ejemplo de política de punto final de VPC

Solo permite solicitudes de una dirección IP específica en la VPC especificada

En el siguiente ejemplo, se muestra una política que solo permite que las solicitudes procedentes de una dirección IP específica en la VPC especificada se ejecuten correctamente. Las solicitudes de otras direcciones IP devolverán un error.

```
{
    "Statement": [
        {
            "Action": "kafkaconnect:*",
            "Effect": "Allow",
            "Principal": "*",
            "Resource": "*",
            "Condition": {
                "IpAddress": {
                    "aws:VpcSourceIp": "192.0.2.123"
                },
        "StringEquals": {
                    "aws:SourceVpc": "vpc-555555555555"
                }
            }
        }
    ]
}
```

Autenticación y autorización para Amazon MSK APIs

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los recursos. AWS Los administradores de IAM controlan quién se puede autenticar (iniciar sesión) y autorizar (tener permisos) para utilizar los recursos de Amazon MSK. La IAM es una Servicio de AWS herramienta que puede utilizar sin coste adicional.

Temas

- Cómo funciona Amazon MSK con IAM

- Ejemplos de políticas de Amazon MSK basadas en identidades
- Roles vinculados a servicios para Amazon MSK
- AWS políticas gestionadas para Amazon MSK
- Solución de problemas de identidad y acceso de Amazon MSK

Cómo funciona Amazon MSK con IAM

Antes de utilizar IAM para administrar el acceso a Amazon MSK, debe conocer qué características de IAM se encuentran disponibles con Amazon MSK. Para obtener una visión general de cómo Amazon MSK y otros AWS servicios funcionan con IAM, consulte AWS Servicios que funcionan con IAM en la Guía del usuario de IAM.

Temas

- Políticas basadas en identidades de Amazon MSK
- Políticas basadas en recursos de Amazon MSK
- Autorización basada en etiquetas de Amazon MSK
- Roles de IAM de Amazon MSK

Políticas basadas en identidades de Amazon MSK

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. Amazon MSK admite acciones, claves de condiciones y recursos específicos. Para obtener información sobre todos los elementos que utiliza en una política JSON, consulte Referencia de los elementos de las políticas JSON de IAM en la Guía del usuario de IAM.

Acciones para las políticas basadas en identidades de Amazon MSK

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Las acciones de políticas de Amazon MSK utilizan el siguiente prefijo antes de la acción: `kafka:`. Por ejemplo, para conceder a alguien permiso para describir un clúster de MSK con la operación de la API `DescribeCluster` de Amazon MSK, incluya la acción `kafka:DescribeCluster` en su política. Las instrucciones de la política deben incluir un elemento `Action` o un elemento `NotAction`. Amazon MSK define su propio conjunto de acciones que describen las tareas que se pueden realizar con este servicio.

Para especificar varias acciones en una única instrucción, sepárelas con comas del siguiente modo:

```
"Action": ["kafka:action1", "kafka:action2"]
```

Puede utilizar caracteres comodín para especificar varias acciones (*). Por ejemplo, para especificar todas las acciones que comiencen con la palabra `Describe`, incluya la siguiente acción:

```
"Action": "kafka:Describe*"
```

Para consultar una lista de acciones de Amazon MSK, consulte Acciones, recursos y claves de condición para Amazon Managed Streaming para Apache Kafka en la Guía del usuario de IAM.

Recursos para las políticas basadas en identidades de Amazon MSK

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puedes hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utiliza un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

El recurso de instancia de Amazon MSK tiene el siguiente ARN:

```
arn:${Partition}:kafka:${Region}:${Account}:cluster/${ClusterName}/${UUID}
```

Para obtener más información sobre el formato de ARNs, consulte Amazon Resource Names (ARNs) y AWS Service Namespaces.

Por ejemplo, para especificar la instancia de `CustomerMessages` en su instrucción, utilice el siguiente ARN:

```
"Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/CustomerMessages/abcd1234-abcd-dcba-4321-a1b2abcd9f9f-2"
```

Para especificar todas las instancias que pertenecen a una cuenta específica, utilice el carácter comodín (*):

```
"Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/*"
```

Algunas acciones de Amazon MSK, como las que se utilizan para crear recursos, no se pueden llevar a cabo en un recurso específico. En dichos casos, debe utilizar el carácter comodín (*).

```
"Resource": "*"
```

Para especificar varios recursos en una sola sentencia, sepárelos ARNs con comas.

```
"Resource": ["resource1", "resource2"]
```

Para ver una lista de los tipos de recursos de Amazon MSK y sus tipos ARNs, consulte Recursos definidos por Amazon Managed Streaming for Apache Kafka Kafka en la Guía del usuario de IAM. Para obtener información acerca de las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por Amazon Managed Streaming para Apache Kafka.

Claves de condición para las políticas basadas en identidades de Amazon MSK

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puedes crear expresiones

condicionales que utilizan operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puedes utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puedes conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulta Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Amazon MSK define su propio conjunto de claves de condición y también admite el uso de algunas claves de condición globales. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Para consultar una lista de claves de condición de Amazon MSK, consulte Claves de condición de Amazon Managed Streaming para Apache Kafka en la Guía del usuario de IAM. Para obtener más información acerca de las acciones y los recursos con los que puede utilizar una clave de condición, consulte Acciones definidas por Amazon Managed Streaming para Apache Kafka.

Ejemplos de políticas basadas en identidades de Amazon MSK

Para ver ejemplos de políticas basadas en identidades de Amazon MSK, consulte Ejemplos de políticas de Amazon MSK basadas en identidades.

Políticas basadas en recursos de Amazon MSK

Amazon MSK admite una política de clústeres (también conocida como política basada en recursos) para su uso con los clústeres de Amazon MSK. Puede utilizar una política de clústeres para definir qué entidades principales de IAM tienen permisos entre cuentas para configurar la conectividad privada con su clúster de Amazon MSK. Si se utiliza con la autenticación de clientes de IAM, también puede utilizar la política de clústeres para definir de forma pormenorizada los permisos del plano de datos de Kafka para los clientes que se conectan.

Para ver un ejemplo de cómo configurar una política de clúster, consulte Paso 2: asociación de una política de clúster al clúster de MSK.

Autorización basada en etiquetas de Amazon MSK

Puede asociar etiquetas a clústeres de Amazon MSK. Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `kafka:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`. Para obtener información sobre el etiquetado de los recursos de Amazon MSK, consulte. the section called “Etiquetar un clúster”

Solo puede controlar el acceso al clúster con la ayuda de etiquetas. Para etiquetar temas y grupos de consumidores, debe añadir una declaración independiente en sus políticas sin etiquetas.

Para ver un ejemplo de una política basada en la identidad para limitar el acceso a un clúster en función de las etiquetas de ese clúster, consulte. Acceso a los clústeres de Amazon MSK basados en etiquetas

Puede utilizar condiciones en la política basada en identidades para controlar el acceso a los recursos de Amazon MSK basados en etiquetas. El siguiente ejemplo muestra una política que permite al usuario describir el clúster, obtener sus agentes de arranque, enumerar sus nodos de agentes, actualizarlo y eliminarlo. Sin embargo, esta política solo concede el permiso si la etiqueta de clúster `Owner` tiene el valor de ese usuario. `username` La segunda declaración de la siguiente política permite el acceso a los temas del clúster. La primera declaración de esta política no autoriza el acceso a ningún tema.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AccessClusterIfOwner",
      "Effect": "Allow",
      "Action": [
        "kafka:Describe*",
        "kafka:Get*",
        "kafka:List*",
        "kafka:Update*",
        "kafka:Delete*"
      ],
      "Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/*",
      "Condition": {
```

```
        "StringEquals": {
          "aws:ResourceTag/Owner": "${aws:username}"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "kafka-cluster:*Topic*",
        "kafka-cluster:WriteData",
        "kafka-cluster:ReadData"
      ],
      "Resource": [
        "arn:aws:kafka:us-east-1:123456789012:topic/*"
      ]
    }
  ]
}
```

Roles de IAM de Amazon MSK

Un rol de IAM es una entidad de la cuenta de Amazon Web Services que dispone de permisos específicos.

Uso de credenciales temporales con Amazon MSK

Puede utilizar credenciales temporales para iniciar sesión con federación, asumir un rol de IAM o asumir un rol de acceso entre cuentas. Las credenciales de seguridad temporales se obtienen llamando a operaciones de AWS STS API como AssumeRoleo GetFederationToken.

Amazon MSK admite el uso de credenciales temporales.

Roles vinculados a servicios

Los roles vinculados a servicios permiten que Amazon Web Services acceda a los recursos de otros servicios para completar una acción en su nombre. Los roles vinculados a servicios aparecen en la cuenta de IAM y son propiedad del servicio. Un administrador puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Amazon MSK admite roles vinculados a servicios. Para obtener más información sobre cómo crear o administrar roles vinculados a servicios de Amazon MSK, consulte the section called “Roles vinculados a servicios”.

Ejemplos de políticas de Amazon MSK basadas en identidades

De forma predeterminada, los usuarios y roles de IAM no tienen permiso para llevar a cabo operaciones de la API de Amazon MSK. Un administrador debe crear políticas de IAM que concedan permisos a los usuarios y a los roles para realizar operaciones de la API concretas en los recursos especificados que necesiten. El administrador debe adjuntar esas políticas a los usuarios o grupos de IAM que necesiten esos permisos.

Para obtener información acerca de cómo crear una política basada en identidad de IAM con estos documentos de políticas JSON de ejemplo, consulte Creación de políticas en la pestaña JSON en la Guía del usuario de IAM.

Temas

- Prácticas recomendadas relativas a políticas
- Cómo permitir a los usuarios consultar sus propios permisos
- Acceso a un clúster de Amazon MSK
- Acceso a los clústeres de Amazon MSK basados en etiquetas

Prácticas recomendadas relativas a políticas

Las políticas basadas en identidades determinan si alguien puede crear, eliminar o acceder a los recursos de Amazon MSK de la cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos en muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulta las políticas administradas por AWS o las políticas administradas por AWS para funciones de tarea en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulta Políticas y permisos en IAM en la Guía del usuario de IAM.

- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulta Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Validación de políticas con el Analizador de acceso de IAM en la Guía del usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte Acceso seguro a la API con MFA en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas gestionadas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API o. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
```

```
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Acceso a un clúster de Amazon MSK

En este ejemplo, desea conceder acceso a un usuario de IAM de su cuenta de Amazon Web Services a uno de sus clústeres, `purchaseQueriesCluster`. Esta directiva permite al usuario describir el clúster, obtener sus agentes de arranque, enumerar sus nodos de agentes y actualizarlo.

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

Acceso a los clústeres de Amazon MSK basados en etiquetas

Puede utilizar condiciones en la política basada en identidades para controlar el acceso a los recursos de Amazon MSK basados en etiquetas. En este ejemplo se muestra cómo crear una directiva que permita al usuario describir el clúster, obtener sus agentes de arranque, enumerar sus nodos de agentes, actualizarla y eliminarla. Sin embargo, los permisos solo se conceden si la etiqueta de clúster `Owner` tiene el valor del nombre de usuario de dicho usuario.

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

También puede asociar esta política al usuario de IAM en su cuenta. Si un usuario llamado `richard-roe` intenta actualizar un clúster de MSK, el clúster debe tener la etiqueta `Owner=richard-roe` o `owner=richard-roe`. De lo contrario, se le deniega el acceso. La clave de la etiqueta de condición `Owner` coincide con los nombres de las claves de condición `Owner` y

`owner` porque no distinguen entre mayúsculas y minúsculas. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.

Roles vinculados a servicios para Amazon MSK

Amazon MSK utiliza funciones vinculadas a servicios AWS Identity and Access Management (IAM). Un rol vinculado a servicios es un tipo único de rol de IAM que se encuentra vinculado directamente a Amazon MSK. Amazon MSK predefine las funciones vinculadas al servicio e incluyen todos los permisos que el servicio requiere para llamar a otros AWS servicios en su nombre.

Un rol vinculado a servicios simplifica la configuración de Amazon MSK porque ya no tendrá que agregar de forma manual los permisos necesarios. Amazon MSK define los permisos de sus roles vinculados a servicios. A menos que se defina lo contrario, solo Amazon MSK puede asumir sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda asociar a ninguna otra entidad de IAM.

Para obtener información acerca de otros servicios que admiten roles vinculados a servicios, consulte Servicios de Amazon Web Services que funcionan con IAM y busque los servicios que tengan Sí en la columna Roles vinculados a servicios. Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado a servicios en cuestión.

Temas

- Permisos de roles vinculados a servicios para Amazon MSK
- Creación de un rol vinculado a los servicios para Amazon MSK
- Edición de un rol vinculado a los servicios para Amazon MSK
- Regiones admitidas para los roles vinculados a servicios de Amazon MSK

Permisos de roles vinculados a servicios para Amazon MSK

Amazon MSK usa el rol vinculado a servicios denominado AWSServiceRoleForKafka. Amazon MSK utiliza esta función para acceder a sus recursos y realizar operaciones como las siguientes:

- `*NetworkInterface`: cree y administre interfaces de red en la cuenta del cliente que hagan que los clientes de la VPC del cliente puedan acceder a los agentes de clústeres.
- `*VpcEndpoints`— administre los puntos finales de la VPC en la cuenta del cliente para que los agentes de clústeres sean accesibles a los clientes de la VPC del cliente que utilizan. AWS PrivateLink Amazon MSK usa permisos para `DescribeVpcEndpoints`, `ModifyVpcEndpoint` y `DeleteVpcEndpoints`.

- `secretsmanager`— gestione las credenciales de los clientes con. AWS Secrets Manager
- `GetCertificateAuthorityCertificate`: recupere el certificado para su autoridad de certificación privada.

Este rol vinculado a un servicio se adjunta a la siguiente política administrada: `KafkaServiceRolePolicy`. Para obtener actualizaciones de esta política, consulte KafkaServiceRolePolicy.

El rol vinculado al servicio AWSServiceRoleForKafka depende de los siguientes servicios para asumir el rol:

- `kafka.amazonaws.com`

La política de permisos del rol permite que Amazon MSK realice las siguientes acciones en los recursos.

```
{
 "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_*"
    }
   }
  }
 ]
}
```

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o rol) crear, editar o eliminar un rol vinculado a servicios. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Creación de un rol vinculado a los servicios para Amazon MSK

No necesita crear manualmente un rol vinculado a un servicio. Cuando crea un clúster de Amazon MSK en la AWS Management Console, la o la AWS API AWS CLI, Amazon MSK crea el rol vinculado al servicio por usted.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Al crear un clúster de Amazon MSK, Amazon MSK se encarga de crear de nuevo el rol vinculado a servicios en su nombre.

Edición de un rol vinculado a los servicios para Amazon MSK

Amazon MSK no permite editar el rol vinculado a servicios de AWSServiceRoleForKafka. Después de crear un rol vinculado al servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM. Para obtener más información, consulte Edición de un rol vinculado a servicios en la Guía del usuario de IAM.

Regiones admitidas para los roles vinculados a servicios de Amazon MSK

Amazon MSK admite el uso de roles vinculados a servicios en todas las regiones en las que se encuentra disponible el servicio. Para obtener más información, consulte AWS Regiones y puntos de conexión.

AWS políticas gestionadas para Amazon MSK

Una política AWS administrada es una política independiente creada y administrada por. AWS AWS Las políticas administradas están diseñadas para proporcionar permisos para muchos casos de uso comunes, de modo que pueda empezar a asignar permisos a usuarios, grupos y funciones.

Ten en cuenta que es posible que las políticas AWS administradas no otorguen permisos con privilegios mínimos para tus casos de uso específicos, ya que están disponibles para que los usen todos los AWS clientes. Se recomienda definir políticas administradas por el cliente específicas para sus casos de uso a fin de reducir aún más los permisos.

No puedes cambiar los permisos definidos en AWS las políticas administradas. Si AWS actualiza los permisos definidos en una política AWS administrada, la actualización afecta a todas las identidades principales (usuarios, grupos y roles) a las que está asociada la política. AWS es más probable que actualice una política AWS administrada cuando Servicio de AWS se lance una nueva o cuando estén disponibles nuevas operaciones de API para los servicios existentes.

Para obtener más información, consulte Políticas administradas de AWS en la Guía del usuario de IAM.

AWS política gestionada: Amazon MSKFull Access

Esta política otorga permisos administrativos que brindan a una entidad principal acceso completo a todas las acciones de Amazon MSK. Los permisos de esta política se agrupan de la siguiente manera:

- Los permisos de Amazon MSK permiten todas las acciones de Amazon MSK.
- Permisos de **`Amazon EC2`**: en esta política, son obligatorios para validar los recursos transferidos en una solicitud de API. Esto es para garantizar que Amazon MSK pueda utilizar correctamente los recursos con un clúster. El resto de los EC2 permisos de Amazon de esta política permiten a Amazon MSK crear AWS los recursos necesarios para que pueda conectarse a sus clústeres.
- Permisos de **`AWS KMS`**: se utilizan durante las llamadas a la API para validar los recursos transferidos en una solicitud. Son necesarios para que Amazon MSK pueda utilizar la clave pasada con el clúster de Amazon MSK.
- Permisos de **`CloudWatch Logs, Amazon S3, and Amazon Data Firehose`**: son obligatorios para que Amazon MSK garantice que se pueda acceder a los destinos de entrega de registros y que sean válidos para el uso de los registros de los agentes.
- Permisos de **`IAM`**: son obligatorios para que Amazon MSK pueda crear un rol vinculado a los servicios en su cuenta y para permitirle transferir un rol de ejecución de servicios a Amazon MSK.

```
{
 "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 política gestionada: Amazon MSKRead OnlyAccess

Esta política otorga permisos de solo lectura que permiten a los usuarios ver información en Amazon MSK. Las entidades principales con esta política asociada no pueden realizar actualizaciones ni eliminar los recursos existentes, ni pueden crear nuevos recursos de Amazon MSK. Por ejemplo, las entidades principales con estos permisos pueden ver la lista de configuraciones y clústeres asociados a su cuenta, pero no pueden cambiar la configuración ni los ajustes de ningún clúster. Los permisos de esta política se agrupan de la siguiente manera:

- Permisos de **`Amazon MSK`**: permiten enumerar los recursos de Amazon MSK, describirlos y obtener información sobre ellos.
- **`Amazon EC2`**permisos: se utilizan para describir la VPC de Amazon, las subredes y ENIs los grupos de seguridad asociados a un clúster.
- Permisos de **`AWS KMS`**: se utiliza para describir la clave asociada al clúster.

```
{
    "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 política gestionada: KafkaServiceRolePolicy

No puede adjuntarse KafkaServiceRolePolicy a sus entidades de IAM. Esta política se encuentra asociada a un rol vinculado a servicios que permite a Amazon MSK realizar acciones como administrar puntos de conexión (conectores) de VPC en clústeres de MSK, administrar interfaces de red y administrar credenciales de clúster con AWS Secrets Manager. Para obtener más información, consulte the section called “Roles vinculados a servicios”.

AWS política gestionada: AWSMSKReplicator ExecutionRole

La política `AWSMSKReplicatorExecutionRole` concede permisos al Replicador Amazon MSK para replicar datos entre los clústeres de MSK. Los permisos de esta política se agrupan de la siguiente manera:

- **`cluster`**: otorga al Replicador Amazon MSK permisos para conectarse al clúster con la autenticación de IAM. También concede permisos para describir y modificar el clúster.
- **`topic`**: otorga al Replicador Amazon MSK permisos para describir, crear y modificar un tema, así como para modificar la configuración dinámica del tema.
- **`consumer group`**: otorga al Replicador Amazon MSK permisos para describir y modificar grupos de consumidores, leer y escribir datos de un clúster de MSK y eliminar temas internos que creó el Replicador.

```
{
 "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 actualiza las políticas AWS gestionadas

Consulta los detalles sobre las actualizaciones de las políticas AWS gestionadas de Amazon MSK desde que este servicio comenzó a realizar el seguimiento de estos cambios.

Cambio	Descripción	Fecha
WriteDataIdempotently permiso agregado a AWSMSKReplicator ExecutionRole: actualización a una política existente	Amazon MSK ha añadido WriteDataIdempotently permisos a la AWSMSKReplicator ExecutionRole política para admitir la replicación de datos entre clústeres de MSK.	12 de marzo de 2024
AWSMSKReplicatorExecutionRole: política nueva	Amazon MSK agregó una AWSMSKReplicator Execution Role política para admitir Amazon MSK Replicator.	4 de diciembre de 2023
Amazon MSKFull Access: actualización de una política existente	Amazon MSK agregó permisos para admitir el Replicador MSK de Amazon.	28 de septiembre de 2023
KafkaServiceRolePolicy: actualización de una política actual	Amazon MSK agregó permisos para admitir la conectividad privada con varias VPC.	8 de marzo de 2023
Amazon MSKFull Access: actualización de una política existente	Amazon MSK ha añadido nuevos EC2 permisos de	30 de noviembre de 2021

Cambio	Descripción	Fecha
	Amazon para poder conectars e a un clúster.	
Amazon MSKFull Access: actualización de una política existente	Amazon MSK agregó un nuevo permiso que le permite describir las tablas de EC2 rutas de Amazon.	19 de noviembre de 2021
Amazon MSK comenzó a hacer un seguimiento de los cambios	Amazon MSK comenzó a realizar un seguimiento de los cambios en sus políticas AWS gestionadas.	19 de noviembre de 2021

Solución de problemas de identidad y acceso de Amazon MSK

Utilice la siguiente información para diagnosticar y solucionar los problemas habituales que pueden surgir cuando se trabaja con Amazon MSK e IAM.

Temas

- No tengo autorización para realizar una acción en Amazon MSK

No tengo autorización para realizar una acción en Amazon MSK

Si AWS Management Console le indica que no está autorizado a realizar una acción, debe ponerse en contacto con su administrador para obtener ayuda. El gestionador es la persona que le proporcionó las credenciales de inicio de sesión.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para eliminar un clúster, pero no tiene permisos `kafka:DeleteCluster`.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 kafka:DeleteCluster on resource: purchaseQueriesCluster
```

En este caso, Mateo pide a su administrador que actualice sus políticas de forma que pueda obtener acceso al recurso `purchaseQueriesCluster` mediante la acción `kafka:DeleteCluster`.

Autenticación y autorización para Apache Kafka APIs

Puede utilizar IAM para autenticar clientes y permitir o denegar acciones de Apache Kafka. Como alternativa, puede usar TLS o SASL/SCRAM para autenticar clientes y Apache Kafka ACLs para permitir o denegar acciones.

Para obtener información sobre cómo controlar quién puede realizar las operaciones de Amazon MSK en su clúster, consulte the section called “Autenticación y autorización para Amazon MSK APIs”.

Temas

- Control de acceso de IAM
- Autenticación TLS mutua de clientes para Amazon MSK
- Autenticación de credenciales de inicio de sesión con AWS Secrets Manager
- Apache Kafka ACLs

Control de acceso de IAM

El control de acceso de IAM para Amazon MSK le permite gestionar tanto la autenticación como la autorización de su clúster de MSK. Esto elimina la necesidad de utilizar un mecanismo de autenticación y otro mecanismo de autorización. Por ejemplo, cuando un cliente intenta escribir en su clúster, Amazon MSK utiliza IAM para verificar si el cliente es una identidad autenticada y si está autorizado para producir en su clúster.

El control de acceso de IAM funciona para clientes Java y no Java, incluidos los clientes de Kafka escritos en Python JavaScript, Go y.NET. El control de acceso de IAM para clientes que no son de Java está disponible para los clústeres de MSK con la versión 2.7.1 o superior de Kafka.

Para hacer posible el control de acceso de IAM, Amazon MSK realiza pequeñas modificaciones en el código fuente de Apache Kafka. Estas modificaciones no supondrán una diferencia notable en su experiencia con Apache Kafka. Amazon MSK registra los eventos de acceso para que pueda auditarlos.

Puede invocar la ACL de Apache Kafka para un clúster de MSK que utilice APIs el control de acceso de IAM. Sin embargo, Apache Kafka no afecta a la ACLs autorización de las identidades de IAM. Debe utilizar las políticas de IAM para controlar el acceso de las identidades de IAM.

⚠ Consideraciones importantes

Cuando utilice el control de acceso de IAM con su clúster de MSK, tenga en cuenta las siguientes consideraciones importantes:

- El control de acceso de IAM no se aplica a los nodos de Apache. ZooKeeper Para obtener más información sobre cómo puede controlar el acceso a estos nodos, consulte Controle el acceso a ZooKeeper los nodos de Apache en su clúster de Amazon MSK.
- La configuración `allow.everyone.if.no.acl.found` de Apache Kafka no tiene efecto si el clúster utiliza el control de acceso de IAM.
- Puede invocar la ACL de Apache Kafka APIs para un clúster de MSK que utilice el control de acceso de IAM. Sin embargo, Apache Kafka no afecta a la ACLs autorización de las identidades de IAM. Debe utilizar las políticas de IAM para controlar el acceso de las identidades de IAM.

Cómo funciona el control de acceso de IAM para Amazon MSK

Para utilizar el control de acceso de IAM para Amazon MSK, lleve a cabo los pasos que se describen en detalle en los siguientes temas:

- Creación de un clúster de Amazon MSK que utilice el control de acceso de IAM
- Configuración de clientes para el control de acceso de IAM
- Creación de políticas de autorización para el rol de IAM
- Obtener los agentes de arranque para el control de acceso de IAM

Creación de un clúster de Amazon MSK que utilice el control de acceso de IAM

En esta sección se explica cómo puede utilizar la AWS Management Console API o la AWS CLI para crear un clúster de Amazon MSK que utilice el control de acceso de IAM. Para obtener información sobre cómo activar el control de acceso de IAM en un clúster existente, consulte Actualización de la configuración de seguridad de un clúster de Amazon MSK.

Úselo AWS Management Console para crear un clúster que utilice el control de acceso de IAM

1. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/.
2. Elija Create cluster.

3. Elija Crear clúster con configuración personalizada.
4. En la sección Autenticación, elija Control de acceso de IAM.
5. Complete el resto del flujo de trabajo para crear un clúster.

Utilice la API o la AWS CLI para crear un clúster que utilice el control de acceso de IAM

- Para crear un clúster con el control de acceso de IAM habilitado, utilice la CreateClusterAPI o el comando CLI create-cluster y pase el siguiente JSON para `ClientAuthentication` el parámetro:. `"ClientAuthentication": { "Sasl": { "Iam": { "Enabled": true } }`

Configuración de clientes para el control de acceso de IAM

Para permitir que los clientes se comuniquen con un clúster de MSK que utiliza el control de acceso de IAM, elija uno de estos mecanismos:

- Configuración de clientes que no son de Java mediante un mecanismo SASL_OAUTHBEARER
- Configuración del cliente Java mediante SASL_OAUTHBEARER un mecanismo o AWS_MSK_IAM mecanismo

Utilice el SASL_OAUTHBEARER mecanismo para configurar IAM

1. Edite el archivo de configuración client.properties con el siguiente ejemplo de cliente de Python Kafka. Los cambios de configuración son parecidos en otros idiomas.

```
from kafka import KafkaProducer
from kafka.errors import KafkaError
from kafka.sasl.oauth import AbstractTokenProvider
import socket
import time
from aws_msk_iam_sasl_signer import MSKAuthTokenProvider

class MSKTokenProvider():
    def token(self):
        token, _ = MSKAuthTokenProvider.generate_auth_token('<my Región de AWS>')
        return token

tp = MSKTokenProvider()
```

```
producer = KafkaProducer(
    bootstrap_servers='<myBootstrapString>',
    security_protocol='SASL_SSL',
    sasl_mechanism='OAUTHBEARER',
    sasl_oauth_token_provider=tp,
    client_id=socket.gethostname(),
)

topic = "<my-topic>"
while True:
    try:
        inp=input(">")
        producer.send(topic, inp.encode())
        producer.flush()
        print("Produced!")
    except Exception:
        print("Failed to send message:", e)

producer.close()
```

2. Descargue la biblioteca auxiliar para el idioma de configuración que elija y siga las instrucciones de la sección Primeros pasos de la página de inicio de esa biblioteca de idiomas.

   - 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: -signer-go #getting -started https://github.com/aws/ aws-msk-iam-sasl
   - .NET: -signer-net #getting -started https://github.com/aws/ aws-msk-iam-sasl
   - JAVA: el SASL_OAUTHBEARER soporte para Java está disponible a través del archivo jar `aws-msk-iam-auth`

Utilice el AWS_MSK_IAM mecanismo personalizado de MSK para configurar IAM

1. Agregue la línea siguiente al archivo `client.properties`. *<PATH_TO_TRUST_STORE_FILE>*Sustitúyala por la ruta completa al archivo del almacén de confianza del cliente.

   > **Note**
   >
   > Si no desea utilizar un certificado específico, puede eliminar `ssl.truststore.location=`*<PATH_TO_TRUST_STORE_FILE>*

del archivo `client.properties`. Cuando no especifica un valor para `ssl.truststore.location`, el proceso de Java utiliza el certificado predeterminado.

```
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
```

Para usar un perfil con nombre que haya creado para AWS las credenciales, inclúyalo `awsProfileName="your profile name";` en el archivo de configuración del cliente. Para obtener información sobre los perfiles con nombre, consulte Perfiles con nombre en la AWS CLI documentación.

2. Descargue el último archivo aws-msk-iam-authJAR estable y colóquelo en la ruta de clases. Si usa Maven, agregue la siguiente dependencia y ajuste el número de versión según sea necesario:

```
<dependency>
    <groupId>software.amazon.msk</groupId>
    <artifactId>aws-msk-iam-auth</artifactId>
    <version>1.0.0</version>
</dependency>
```

El complemento de cliente de Amazon MSK es de código abierto y cuenta con la licencia Apache 2.0.

Creación de políticas de autorización para el rol de IAM

Asocie una política de autorización al rol de IAM que corresponda al cliente. En una política de autorización, se especifican las acciones que se van a permitir o denegar para el rol. Si tu cliente está en una EC2 instancia de Amazon, asocia la política de autorización al rol de IAM de esa EC2 instancia de Amazon. Como alternativa, puede configurar su cliente para que utilice un perfil con nombre y, a continuación, asociar la política de autorización al rol de ese perfil con nombre. Configuración de clientes para el control de acceso de IAM describe cómo configurar un cliente para utilizar un perfil con nombre asignado.

Para obtener más información sobre cómo crear una política de IAM, consulte Creación de políticas de IAM.

El siguiente es un ejemplo de política de autorización para un clúster denominado MyTestCluster. Para entender la semántica de los elementos `Action` y `Resource`, consulte Semántica de las acciones y los recursos de la política de autorización de IAM.

> **Important**
>
> Los cambios que realices en una política de IAM se reflejan en el IAM APIs y de forma inmediata. AWS CLI Sin embargo, puede pasar un tiempo considerable hasta que el cambio en la política surta efecto. En la mayoría de los casos, los cambios en la política entran en vigor en menos de un minuto. En ocasiones, las condiciones de la red pueden aumentar la demora.

```
{
    "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-1:123456789012:topic/MyTestCluster/*"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeGroup"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:123456789012:group/MyTestCluster/*"
            ]
        }
    ]
}
```

Para obtener información sobre cómo crear una política con elementos de acción que se correspondan con los casos de uso habituales de Apache Kafka, como la producción y el consumo de datos, consulte Casos de uso habituales de la política de autorización de clientes.

En las versiones 2.8.0 y posteriores de Kafka, el WriteDataIdempotentlypermiso está obsoleto (KIP-679). Se utiliza `enable.idempotence = true` de forma predeterminada. Por lo tanto, para las versiones 2.8.0 y posteriores de Kafka, IAM no ofrece la misma funcionalidad que Kafka. ACLs No es posible acceder `WriteDataIdempotently` a un tema solo proporcionando `WriteData` acceso a ese tema. Esto no afecta al caso cuando `WriteData` se proporciona a TODOS los temas. En ese caso, `WriteDataIdempotently` está permitido. Esto se debe a las diferencias en la implementación de la lógica de IAM y en la forma en que ACLs se implementa el Kafka. Además, escribir sobre un tema de manera despotente también requiere acceso a. `transactional-ids`

Para solucionar este problema, te recomendamos que utilices una política similar a la siguiente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:AlterCluster",
                "kafka-cluster:DescribeCluster",
                "kafka-cluster:WriteDataIdempotently"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:123456789012:cluster/MyTestCluster/
abcd1234-0123-abcd-5678-1234abcd-1"
            ]
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:*Topic*",
                "kafka-cluster:WriteData",
                "kafka-cluster:ReadData"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:123456789012:topic/MyTestCluster/
abcd1234-0123-abcd-5678-1234abcd-1/TestTopic",
                "arn:aws:kafka:us-east-1:123456789012:transactional-id/MyTestCluster/
abcd1234-0123-abcd-5678-1234abcd-1/*"
            ]
        }
    ]
}
```

En este caso, `WriteData` permite escribir en `TestTopic` y, al mismo tiempo, `WriteDataIdempotently` permite escrituras idempotentes en el clúster. Esta política también añade el acceso a los `transactional-id` recursos que se necesitarán.

Como `WriteDataIdempotently` se trata de un permiso a nivel de clúster, no puede usarlo a nivel de tema. Si `WriteDataIdempotently` está restringido al nivel de tema, esta política no funcionará.

Obtener los agentes de arranque para el control de acceso de IAM

Consulte Obtención de agentes de arranque para un clúster de Amazon MSK.

Semántica de las acciones y los recursos de la política de autorización de IAM

Actualmente, el control de acceso de IAM para Amazon MSK no admite las acciones de clúster internas de Kafka. Esto incluye la WriteTxnMarkers API, que Kafka utiliza para finalizar las transacciones. Para finalizar las transacciones, le recomendamos que utilice la autenticación SCRAM o mTLS con la autenticación adecuada ACLs en lugar de la autenticación de IAM.

En esta sección se explica la semántica de los elementos de acción y recursos que puede utilizar en una política de autorización de IAM. Para ver una política de ejemplo, consulte Creación de políticas de autorización para el rol de IAM.

Acciones de la política de autorización

En la siguiente tabla se enumeran las acciones que puede incluir en una política de autorización cuando utiliza el control de acceso de IAM para Amazon MSK. Si incluye en su política de

autorización una acción de la columna Acción de la tabla, también debe incluir las acciones correspondientes de la columna Acciones obligatorias.

Acción	Descripción	Acciones obligatorias	Recursos necesarios de	Aplicable a los clústeres sin servidor
`kafka-cluster:Connect`	Otorga permiso para conectarse y autenticarse en el clúster.	Ninguno	clúster	Sí
`kafka-cluster:DescribeCluster`	Otorga permiso para describir varios aspectos del clúster, equivalente a la ACL DESCRIBE CLUSTER de Apache Kafka.	`kafka-cluster:Connect`	clúster	Sí

`kafka-cluster:AlterCluster`	Otorga permiso para alterar varios aspectos del clúster, equivalente a la ACL ALTER CLUSTER de Apache Kafka.	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeCluster`
			clúster	No

`kafka-cluster:DescribeClusterDynamicConfiguration`	Otorga permiso para describir la configuración dinámica de un clúster, equivalente a la ACL de CLÚSTER	`kafka-cluster:Connect`	clúster	No

Acción	Descripción	Acciones obligatorias	Recursos necesarios de	Aplicable a los clústeres sin servidor
	DESCRIBE_ CONFIGS de Apache Kafka.			

`kafka-cluster:AlterClusterDynamicConfiguration`	Otorga permiso para modificar la configuración dinámica de un clúster, equivalente a la ACL de CLÚSTER ALTER_CONFIGS de Apache Kafka.	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeClusterDynamicConfiguration`
			clúster	No

`kafka-cluster:WriteDataIdempotently`	Otorga permiso para escribir datos idempotentemente en un clúster, equivalente a la ACL de CLÚSTER IDEMPOTENT_WRITE de Apache Kafka.	`kafka-cluster:Connect`
		
		`kafka-cluster:WriteData`
			clúster	Sí

Acción	Descripción	Acciones obligatorias	Recursos necesarios de	Aplicable a los clústeres sin servidor
`kafka-cluster:CreateTopic`	Otorga permiso para crear temas en un clúster, lo que equivale a la ACL CREATE de Apache Kafka. CLUSTER/TOPIC	`kafka-cluster:Connect`	tema	Sí
`kafka-cluster:DescribeTopic`	Otorga permiso para describir temas en un clúster, equivalente a la ACL DESCRIBE TOPIC de Apache Kafka.	`kafka-cluster:Connect`	tema	Sí
`kafka-cluster:AlterTopic`	Otorga permiso para modificar temas en un clúster, equivalente a la ACL ALTER TOPIC de Apache Kafka.	`kafka-cluster:Connect` `kafka-cluster:DescribeTopic`	tema	Sí

Acción	Descripción	Acciones obligatorias	Recursos necesarios de	Aplicable a los clústeres sin servidor
`kafka-cluster:DeleteTopic`	Otorga permiso para eliminar temas de un clúster, equivalente a la ACL DELETE TOPIC de Apache Kafka.	`kafka-cluster:Connect` `kafka-cluster:DescribeTopic`	tema	Sí
`kafka-cluster:DescribeTopicDynamicConfiguration`	Otorga permiso para describir la configuración dinámica de temas en un clúster, equivalente a la ACL DESCRIBE_CONFIGS TOPIC de Apache Kafka.	`kafka-cluster:Connect`	tema	Sí
`kafka-cluster:AlterTopicDynamicConfiguration`	Otorga permiso para modificar la configuración dinámica de temas en un clúster, equivalente a la ACL de TEMA ALTER_CONFIGS de Apache Kafka.	`kafka-cluster:Connect` `kafka-cluster:DescribeTopicDynamicConfiguration`	tema	Sí

Acción	Descripción	Acciones obligatorias	Recursos necesarios de	Aplicable a los clústeres sin servidor

`kafka-cluster:ReadData`	Otorga permiso para leer datos de temas de un clúster, equivalente a la ACL READ TOPIC de Apache Kafka.	`kafka-cluster:Connect`
		`kafka-cluster:DescribeTopic`
		`kafka-cluster:AlterGroup`
			tema	Sí

`kafka-cluster:WriteData`	Otorga permiso para escribir datos en temas de un clúster, equivalente a la ACL WRITE TOPIC de Apache Kafka	`kafka-cluster:Connect` `kafka-cluster:DescribeTopic`	tema	Sí
`kafka-cluster:DescribeGroup`	Otorga permiso para describir grupos en un clúster, equivalente a la ACL DESCRIBE GROUP de Apache Kafka.	`kafka-cluster:Connect`	grupo	Sí

Acción	Descripción	Acciones obligatorias	Recursos necesarios de	Aplicable a los clústeres sin servidor

`kafka-cluster:AlterGroup`	Otorga permiso para unirse a grupos en un clúster, equivalente a la ACL READ GROUP de Apache Kafka.	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeGroup`
			grupo	Sí

`kafka-cluster:DeleteGroup`	Otorga permiso para eliminar grupos de un clúster, equivalente a la ACL DELETE GROUP de Apache Kafka.	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeGroup`
			grupo	Sí

`kafka-cluster:DescribeTransactionalId`	Otorga permiso para describir las transacciones IDs en un clúster, lo que equivale a la ACL DESCRIBE TRANSACTIONAL_ID de Apache Kafka.	`kafka-cluster:Connect`	transactional-id	Sí

Acción	Descripción	Acciones obligatorias	Recursos necesarios de	Aplicable a los clústeres sin servidor

`kafka-cluster:AlterTransactionalId`	Otorga permiso para modificar las transacciones IDs en un clúster, lo que equivale a la ACL WRITE TRANSACTIONAL_ID de Apache Kafka.	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeTransactionalId`
		
		`kafka-cluster:WriteData`
			transactional-id	Sí

Puede utilizar el carácter comodín asterisco (*) cualquier número de veces en una acción después de los dos puntos. A continuación se muestran algunos ejemplos.

- `kafka-cluster:*Topic` significa `kafka-cluster:CreateTopic`, `kafka-cluster:DescribeTopic`, `kafka-cluster:AlterTopic` y `kafka-cluster:DeleteTopic`. No incluye `kafka-cluster:DescribeTopicDynamicConfiguration` ni `kafka-cluster:AlterTopicDynamicConfiguration`.
- `kafka-cluster:*` se refiere a todos los permisos.

Recursos de la política de autorización

En la siguiente tabla se muestran los cuatro tipos de recursos que puede utilizar en una política de autorización cuando utiliza el control de acceso de IAM para Amazon MSK. Puede obtener el nombre de recurso de Amazon (ARN) del clúster desde AWS Management Console o mediante la DescribeClusterAPI o el comando AWS CLI describe-cluster. A continuación, puede usar el ARN del clúster para crear el tema, el grupo y el ID de transacción. ARNs Para especificar un recurso en una política de autorización, se utiliza el ARN de ese recurso.

Recurso	Formato de ARN
Clúster	arn:aws:kafka: ::cluster//*regionaccount-id cluster-name cluster-uuid*
Tema	arn:aws:kafka: *region* :topic///*account-id cluster-name cluster-uuid topic-name*
Grupo	arn:aws:kafka: *region* ::group///*account-id cluster-name cluster-uuid group-name*
ID de transacción	arn:aws:kafka: *region* ::transactional-id///*account-id cluster-name cluster-uuid transactional-id*

Puede utilizar el comodín asterisco (*) cualquier número de veces en cualquier parte del ARN que venga después de `:cluster/`, `:topic/`, `:group/` y `:transactional-id/`. A continuación, se muestran algunos ejemplos de cómo puede utilizar el carácter comodín asterisco (*) para hacer referencia a varios recursos:

- `arn:aws:kafka:us-east-1:0123456789012:topic/MyTestCluster/*`: todos los temas de cualquier clúster con nombre, independientemente del UUID del clúster. MyTestCluster
- `arn:aws:kafka:us-east-1:0123456789012:topic/MyTestCluster/abcd1234-0123-abcd-5678-1234abcd-1/*_test`: todos los temas cuyo nombre termine por «_test» del clúster cuyo nombre MyTestCluster y UUID sean abcd1234-0123-abcd-5678-1234abcd-1.
- `arn:aws:kafka:us-east-1:0123456789012:transactional-id/MyTestCluster/*/5555abcd-1111-abcd-1234-abcd1234-1`: todas las transacciones cuyo identificador de transacción sea 5555abcd-1111-abcd-1234-abcd1234-1, en todas las versiones de un clúster con el nombre de tu cuenta. MyTestCluster Esto significa que si crea un clúster con el nombre MyTestCluster, lo elimina y, a continuación, crea otro clúster con el mismo nombre, puede usar este ARN de recurso para representar el mismo ID de transacciones en ambos clústeres. Sin embargo, no se puede acceder al clúster eliminado.

Casos de uso habituales de la política de autorización de clientes

En la primera columna de la tabla siguiente se muestran algunos casos de uso habituales. Para autorizar a un cliente a llevar a cabo un caso de uso determinado, incluya las acciones necesarias para ese caso de uso en la política de autorización del cliente y establezca `Effect` en `Allow`.

Para obtener información sobre todas las acciones que forman parte del control de acceso de IAM para Amazon MSK, consulte Semántica de las acciones y los recursos de la política de autorización de IAM.

> **Note**
>
> Las acciones se deniegan de forma predeterminada. Debe permitir de forma explícita todas las acciones que quiera que el cliente pueda realizar.

Caso de uso	Acciones obligatorias
Administrador	`kafka-cluster:*`
Crear un tema	`kafka-cluster:Connect` `kafka-cluster:CreateTopic`

Producir datos	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:WriteData`

Consumir datos	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:DescribeGroup`
	`kafka-cluster:AlterGroup`
	`kafka-cluster:ReadData`

Producir datos de forma idempotente	`kafka-cluster:Connect`

Caso de uso	Acciones obligatorias

	`kafka-cluster:DescribeTopic`
	
	`kafka-cluster:WriteData`
	
	`kafka-cluster:WriteDataIdempotently`

Producir datos de forma transaccional	`kafka-cluster:Connect`
	
	`kafka-cluster:DescribeTopic`
	
	`kafka-cluster:WriteData`
	
	`kafka-cluster:DescribeTransactionalId`
	
	`kafka-cluster:AlterTransactionalId`

Describir la configuración de un clúster	`kafka-cluster:Connect`
	
	`kafka-cluster:DescribeClusterDynamicConfiguration`

Actualizar la configuración de un clúster	`kafka-cluster:Connect`
	
	`kafka-cluster:DescribeClusterDynamicConfiguration`
	
	`kafka-cluster:AlterClusterDynamicConfiguration`

Describir la configuración de un tema	`kafka-cluster:Connect`
	
	`kafka-cluster:DescribeTopicDynamicConfiguration`

Caso de uso	Acciones obligatorias

Actualizar la configuración de un tema	`kafka-cluster:Connect`
	
	`kafka-cluster:DescribeTopicDynamicConfiguration`
	
	`kafka-cluster:AlterTopicDynamicConfiguration`

Modificar un tema	`kafka-cluster:Connect`
	
	`kafka-cluster:DescribeTopic`
	
	`kafka-cluster:AlterTopic`

Autenticación TLS mutua de clientes para Amazon MSK

Puede habilitar la autenticación de clientes con TLS para las conexiones desde sus aplicaciones a sus agentes de Amazon MSK. Para usar la autenticación del cliente, necesita una Autoridad de certificación privada de AWS. Autoridad de certificación privada de AWS Puede estar en el Cuenta de AWS mismo clúster o en una cuenta diferente. Para obtener información sobre Autoridad de certificación privada de AWS s, consulte Creación y administración de un Autoridad de certificación privada de AWS.

> **Note**
>
> La autenticación de TLS no está disponible actualmente en las regiones de Pekín y Ningxia.

Amazon MSK no admite listas de revocación de certificados ()CRLs. Para controlar el acceso a los temas de su clúster o bloquear los certificados comprometidos, utilice Apache Kafka ACLs y AWS los grupos de seguridad. Para obtener información sobre el uso de Apache Kafka ACLs, consulte. the section called “Apache Kafka ACLs”

Este tema contiene las siguientes secciones:

- Creación de un clúster de Amazon MSK compatible con la autenticación de clientes
- Configuración de un cliente para que utilice la autenticación

- Producción y consumo de mensajes con la autenticación

Creación de un clúster de Amazon MSK compatible con la autenticación de clientes

Este procedimiento le muestra cómo habilitar la autenticación de clientes mediante un Autoridad de certificación privada de AWS.

> **Note**
>
> Recomendamos encarecidamente utilizar un TLS independiente Autoridad de certificación privada de AWS para cada clúster de MSK cuando utilice un TLS mutuo para controlar el acceso. De este modo, se asegurará de que los certificados TLS firmados PCAs únicamente se autentiquen con un único clúster de MSK.

1. Cree un archivo denominado `clientauthinfo.json` con el siguiente contenido. *`Private-CA-ARN`*Sustitúyalo por el ARN de su PCA.

```
{
   "Tls": {
       "CertificateAuthorityArnList": ["Private-CA-ARN"]
    }
}
```

2. Cree un archivo llamado `brokernodegroupinfo.json` tal y como se describe en the section called “Cree un clúster de Amazon MSK aprovisionado mediante AWS CLI”.
3. La autenticación del cliente precisa que también habilite el cifrado en tránsito entre clientes y agentes. Cree un archivo denominado `encryptioninfo.json` con el siguiente contenido. *`KMS-Key-ARN`*Sustitúyala por el ARN de la clave KMS. Puede establecer `ClientBroker` en `TLS` o `TLS_PLAINTEXT`.

```
{
   "EncryptionAtRest": {
       "DataVolumeKMSKeyId": "KMS-Key-ARN"
    },
   "EncryptionInTransit": {
        "InCluster": true,
        "ClientBroker": "TLS"
    }
```

```
}
```

Para obtener más información sobre el cifrado, consulte the section called “Cifrado de Amazon MSK”.

4. En una máquina en la que la tengas AWS CLI instalada, ejecuta el siguiente comando para crear un clúster con la autenticación y el cifrado en tránsito habilitados. Guarde el ARN del clúster proporcionado en la respuesta.

```
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
```

Configuración de un cliente para que utilice la autenticación

Este proceso describe cómo configurar una EC2 instancia de Amazon para usarla como cliente para usar la autenticación.

En este proceso, se describe cómo producir y consumir mensajes con la autenticación mediante la creación de un equipo cliente, la creación de un tema y la configuración de los ajustes de seguridad necesarios.

1. Crea una EC2 instancia de Amazon para usarla como máquina cliente. Para simplificar, cree esta instancia en la misma VPC que utilizó para el clúster. Consulte the section called “Creación de un equipo cliente” para ver un ejemplo sobre cómo crear dicho equipo cliente.
2. Cree un tema. Para ver un ejemplo, consulte las instrucciones en the section called “Crear un tema”.
3. En una máquina en la que lo tengas AWS CLI instalado, ejecuta el siguiente comando para obtener los agentes de arranque del clúster. *Cluster-ARN*Sustitúyalo por el ARN de su clúster.

```
aws kafka get-bootstrap-brokers --cluster-arn Cluster-ARN
```

Guarde la cadena asociada a `BootstrapBrokerStringTls` en la respuesta.

4. En la máquina de su cliente, ejecute el siguiente comando para utilizar el almacén de confianza de JVM para crear su almacén de confianza del cliente. Si su ruta de JVM es diferente, ajuste el comando en consecuencia.

```
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. En la máquina de su cliente, ejecute el siguiente comando para crear una clave privada para su cliente. Sustituya *Distinguished-NameExample-Alias*,*Your-Store-Pass*, y por *Your-Key-Pass* las cadenas que prefiera.

```
keytool -genkey -keystore kafka.client.keystore.jks -validity 300 -storepass Your-
Store-Pass -keypass Your-Key-Pass -dname "CN=Distinguished-Name" -alias Example-
Alias -storetype pkcs12 -keyalg rsa
```

6. En la máquina de su cliente, ejecute el siguiente comando para crear una solicitud de certificado con la clave privada que creo en el paso anterior.

```
keytool -keystore kafka.client.keystore.jks -certreq -file client-cert-sign-request
 -alias Example-Alias -storepass Your-Store-Pass -keypass Your-Key-Pass
```

7. Abra el archivo `client-cert-sign-request` y asegúrese de que comience por `-----BEGIN CERTIFICATE REQUEST-----` y termine por `-----END CERTIFICATE REQUEST-----`. Si comienza por `-----BEGIN NEW CERTIFICATE REQUEST-----`, elimine la palabra `NEW` (y el espacio único que le sigue) desde el comienzo hasta el final del archivo.

8. En una máquina en la que lo tenga AWS CLI instalado, ejecute el siguiente comando para firmar la solicitud de certificado. *Private-CA-ARN*Sustitúyalo por el ARN de su PCA. Puede cambiar el valor de la validez si lo desea. Aquí utilizamos 300 como ejemplo.

```
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"
```

Guarde el ARN del certificado proporcionado en la respuesta.

> **Note**
>
> Para recuperar el certificado de cliente, utilice el comando `acm-pca get-certificate` y especifique el ARN del certificado. Para obtener más información, consulte get-certificate en la Referencia de comandos de la AWS CLI .

9. Ejecute el siguiente comando para obtener el certificado Autoridad de certificación privada de AWS firmado por usted. *Certificate-ARN*Sustitúyalo por el ARN que obtuvo de la respuesta al comando anterior.

```
aws acm-pca get-certificate --certificate-authority-arn Private-CA-ARN --
certificate-arn Certificate-ARN
```

10. A partir del resultado de JSON de la ejecución del comando anterior, copie las cadenas asociadas a `Certificate` y `CertificateChain`. Pegue estas dos cadenas en un nuevo archivo llamado signed-certificate-from-acm. Pegue la siguiente cadena asociada a `Certificate` primero, seguido de la cadena asociada a `CertificateChain`. Sustituya los caracteres \n por las nuevas líneas. La siguiente es la estructura del archivo después de pegarle el certificado y la cadena del certificado.

```
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
```

11. Ejecute el siguiente comando en el equipo cliente para agregar este certificado a su almacén de claves para que puede presentarlo al hablar con los agentes de MSK.

```
keytool -keystore kafka.client.keystore.jks -import -file signed-certificate-from-
acm -alias Example-Alias -storepass Your-Store-Pass -keypass Your-Key-Pass
```

12. Cree un archivo denominado `client.properties` con el siguiente contenido. Ajuste las ubicaciones del almacén de confianza y de claves a las rutas en las que guardó `kafka.client.truststore.jks`. Sustituya los *{YOUR KAFKA VERSION}* marcadores de posición por su versión de cliente de Kafka.

```
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
```

Producción y consumo de mensajes con la autenticación

En este proceso, se describe cómo producir y consumir mensajes con la autenticación.

1. Ejecute el siguiente comando para crear un tema. El archivo denominado `client.properties` es el que creó en el procedimiento anterior.

   ```
   <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. Ejecute el siguiente comando para iniciar un productor de la consola. El archivo denominado `client.properties` es el que creó en el procedimiento anterior.

   ```
   <path-to-your-kafka-installation>/bin/kafka-console-producer.sh --bootstrap-
   server BootstrapBroker-String --topic ExampleTopic --producer.config
    client.properties
   ```

3. En una nueva ventana de comandos en la máquina de su cliente, ejecute el siguiente comando para iniciar un consumidor de la consola.

   ```
   <path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
   server BootstrapBroker-String --topic ExampleTopic --consumer.config
    client.properties
   ```

4. Escriba mensajes en la ventana del productor y observe cómo aparecen en la ventana del consumidor.

Autenticación de credenciales de inicio de sesión con AWS Secrets Manager

Puede controlar el acceso a sus clústeres de Amazon MSK mediante credenciales de inicio de sesión que se almacenan y protegen mediante AWS Secrets Manager. El almacenamiento de las credenciales de usuario en Secrets Manager reduce la sobrecarga de la autenticación de clústeres, como la auditoría, la actualización y la rotación de credenciales. Secrets Manager también le permite compartir las credenciales de usuario entre clústeres.

Tras asociar un secreto a un clúster de MSK, MSK sincroniza los datos de las credenciales periódicamente.

Este tema contiene las siguientes secciones:

- Funcionamiento de la autenticación de las credenciales de inicio de sesión
- Configurar la SASL/SCRAM autenticación para un clúster de Amazon MSK
- Uso de los usuarios
- Limitaciones al momento de utilizar los secretos SCRAM

Funcionamiento de la autenticación de las credenciales de inicio de sesión

La autenticación con credenciales de inicio de sesión para Amazon MSK utiliza la autenticación SASL/SCRAM (Simple Authentication and Security Layer/ Salted Challenge Response Mechanism). A fin de configurar la autenticación con credenciales de inicio de sesión para un clúster, debe crear un recurso de secreto en AWS Secrets Manager y asociar las credenciales de inicio de sesión a ese secreto.

SASL/SCRAM se define en RFC 5802. SCRAM utiliza algoritmos hash seguros y no transmite credenciales de inicio de sesión de texto sin formato entre el cliente y el servidor.

> **Note**
>
> Al configurar la SASL/SCRAM autenticación para el clúster, Amazon MSK activa el cifrado TLS para todo el tráfico entre clientes y agentes.

Configurar la SASL/SCRAM autenticación para un clúster de Amazon MSK

Para configurar un secreto en AWS Secrets Manager, sigue el tutorial Cómo crear y recuperar un secreto en la Guía del usuario de AWS Secrets Manager.

Tenga en cuenta los siguientes requisitos al crear un secreto para un clúster de Amazon MSK:

- En el tipo de secreto, elija Otro tipo de secretos (por ejemplo, clave de API).
- El nombre secreto debe comenzar con el prefijo AmazonMSK_.
- Debes usar una AWS KMS clave personalizada existente o crear una nueva AWS KMS clave personalizada para tu secreto. Secrets Manager usa la AWS KMS clave predeterminada para un secreto de forma predeterminada.

> **Important**
>
> Un secreto creado con la AWS KMS clave predeterminada no se puede usar con un clúster de Amazon MSK.

- Los datos de sus credenciales de inicio de sesión deben tener el siguiente formato para poder ingresar pares clave-valor mediante la opción Texto no cifrado.

```
{
  "username": "alice",
  "password": "alice-secret"
}
```

- Registre el valor del ARN (nombre del recurso de Amazon) de su secreto.
-

> **Important**
>
> No puede asociar un secreto de Secrets Manager a un clúster que supere los límites descritos en the section called “ Ajuste el tamaño correcto del clúster: número de particiones por agente”.

- Si utiliza el AWS CLI para crear el secreto, especifique un ID de clave o un ARN para el `kms-key-id` parámetro. No especifique un alias.
- Para asociar el secreto a su clúster, utilice la consola de Amazon MSK o la BatchAssociateScramSecretoperación.

> **Important**
>
> Al asociar un secreto a un clúster, Amazon MSK asocia una política de recursos al secreto que permite al clúster acceder a los valores de los secretos que ha definido y leerlos. No debe modificar esta política de recursos. Si lo hace, puede impedir que el clúster acceda a su secreto. Si realiza algún cambio en la política de recursos de secretos o en la clave de KMS utilizada para el cifrado secreto, asegúrese de volver a asociar los secretos a su clúster de MSK. Esto garantizará que tu clúster pueda seguir accediendo a tu secreto.

El siguiente ejemplo de entrada JSON para la operación `BatchAssociateScramSecret` asocia un secreto a un clúster:

```
{
  "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"
  ]
}
```

Conexión a un clúster con credenciales de inicio de sesión

Después de crear un secreto y asociarlo al clúster, puede conectar el cliente con el clúster. El siguiente procedimiento muestra cómo conectar un cliente a un clúster que utiliza la SASL/SCRAM autenticación. También muestra cómo producir y consumir a partir de un tema de ejemplo.

Temas

- Conexión de un cliente a un clúster mediante SASL/SCRAM autenticación
- Solución de problemas de conexión

Conexión de un cliente a un clúster mediante SASL/SCRAM autenticación

1. Ejecute el siguiente comando en una máquina que AWS CLI esté instalada. *clusterARN*Sustitúyalo por el ARN de su clúster.

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

   En el resultado JSON de este comando, guarda el valor asociado a la cadena nombrada`BootstrapBrokerStringSaslScram`. Utilizará este valor en pasos posteriores.

2. En su equipo cliente, cree un archivo de configuración JAAS que contenga las credenciales de usuario almacenadas en su secreto. Por ejemplo, para el usuario alice, cree un archivo llamado `users_jaas.conf` con el siguiente contenido.

   ```
   KafkaClient {
      org.apache.kafka.common.security.scram.ScramLoginModule required
      username="alice"
      password="alice-secret";
   };
   ```

3. Utilice el siguiente comando para exportar el archivo de configuración de JAAS como parámetro de entorno KAFKA_OPTS.

```
export KAFKA_OPTS=-Djava.security.auth.login.config=<path-to-jaas-file>/
users_jaas.conf
```

4. Cree un archivo denominado `kafka.client.truststore.jks` en el directorio `/tmp`.

5. (Opcional) Utilice el siguiente comando para copiar el archivo del almacén de claves JDK de la `cacerts` carpeta JVM al `kafka.client.truststore.jks` archivo que creó en el paso anterior. *JDKFolder*Sustitúyalo por el nombre de la carpeta JDK de la instancia. Por ejemplo, su carpeta JDK podría llamarse `java-1.8.0-openjdk-1.8.0.201.b09-0.amzn2.x86_64`.

```
cp /usr/lib/jvm/JDKFolder/lib/security/cacerts /tmp/kafka.client.truststore.jks
```

6. En el directorio `bin` de su instalación de Apache Kafka, cree un archivo de propiedades del cliente llamado `client_sasl.properties` con el siguiente contenido. Este archivo define el mecanismo y el protocolo SASL.

```
security.protocol=SASL_SSL
sasl.mechanism=SCRAM-SHA-512
```

7. Para crear un tema de ejemplo, ejecuta el siguiente comando. *BootstrapBrokerStringSaslScram*Sustitúyala por la cadena bootstrap broker que obtuviste en el paso 1 de este tema.

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

8. Para continuar con el tema de ejemplo que ha creado, ejecute el siguiente comando en su equipo cliente. *BootstrapBrokerStringSaslScram*Sustitúyala por la cadena bootstrap broker que recuperaste en el paso 1 de este tema.

```
<path-to-your-kafka-installation>/bin/kafka-console-producer.sh --broker-
list BootstrapBrokerStringSaslScram --topic ExampleTopicName --producer.config
 client_sasl.properties
```

9. Para continuar con el tema que ha creado, ejecute el siguiente comando en el equipo cliente. *BootstrapBrokerStringSaslScram*Sustitúyala por la cadena bootstrap broker que obtuviste en el paso 1 de este tema.

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBrokerStringSaslScram --topic ExampleTopicName --from-beginning --
consumer.config client_sasl.properties
```

Solución de problemas de conexión

Al ejecutar los comandos del cliente de Kafka, es posible que se produzcan errores en la memoria apilada de Java, especialmente al trabajar con temas o conjuntos de datos extensos. Estos errores se producen porque las herramientas de Kafka se ejecutan como aplicaciones Java con una configuración de memoria predeterminada que puede ser insuficiente para la carga de trabajo.

Para resolver `Out of Memory Java Heap` los errores, puede aumentar el tamaño del montón de Java modificando la variable de `KAFKA_OPTS` entorno para incluir la configuración de memoria.

En el siguiente ejemplo, se establece el tamaño máximo del montón en 1 GB (). `-Xmx1G` Puede ajustar este valor en función de la memoria del sistema disponible y de los requisitos.

```
export KAFKA_OPTS="-Djava.security.auth.login.config=<path-to-jaas-file>/
users_jaas.conf -Xmx1G"
```

Si se trata de temas extensos, considere la posibilidad de utilizar parámetros basados en el tiempo o en la compensación en lugar de limitar el uso de la memoria`--from-beginning`:

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBrokerStringSaslScram --topic ExampleTopicName --max-messages 1000 --
consumer.config client_sasl.properties
```

Uso de los usuarios

Creación de usuarios: los usuarios se crean en su secreto como pares clave-valor. Al utilizar la opción Texto no cifrado de la consola de Secrets Manager, debe especificar los datos de las credenciales de inicio de sesión en el siguiente formato:

```
{
```

```
  "username": "alice",
  "password": "alice-secret"
}
```

Revocar el acceso de un usuario: para revocar las credenciales de un usuario para acceder a un clúster, le recomendamos que primero elimine o aplique una ACL en el clúster y, a continuación, desasocie el secreto. Esto se debe a lo siguiente:

- La eliminación de un usuario no cierra las conexiones existentes.
- Los cambios realizados en el secreto tardan hasta 10 minutos en propagarse.

Para obtener más información acerca del uso de ACL con Amazon MSK, consulte Apache Kafka ACLs.

En el caso de los clústeres que utilizan ZooKeeper el modo, se recomienda restringir el acceso a ZooKeeper los nodos para evitar que los usuarios los modifiquen. ACLs Para obtener más información, consulte Controle el acceso a ZooKeeper los nodos de Apache en su clúster de Amazon MSK.

Limitaciones al momento de utilizar los secretos SCRAM

Cuando utilice secretos SCRAM, tenga en cuenta las siguientes limitaciones:

- Amazon MSK solo admite la autenticación SCRAM-SHA-512.
- Un clúster de Amazon MSK puede tener hasta 1000 usuarios.
- Debes usar un AWS KMS key con tu secreto. No puede usar un secreto que utilice la clave de cifrado predeterminada de Secrets Manager con Amazon MSK. Para obtener información sobre cómo crear una clave de KMS, consulte Creating symmetric encryption KMS keys.
- No puede utilizar una clave de KMS asimétrica con Secrets Manager.
- Puedes asociar hasta 10 secretos a un clúster a la vez mediante la BatchAssociateScramSecretoperación.
- El nombre de los secretos asociados a un clúster de Amazon MSK debe tener el prefijo AmazonMSK_.
- Los secretos asociados a un clúster de Amazon MSK deben estar en la misma cuenta y AWS región de Amazon Web Services que el clúster.

Apache Kafka ACLs

Apache Kafka tiene un autorizador conectable y viene con una implementación de autorizador. out-of-box Amazon MSK habilita este autorizador en el archivo `server.properties` de los agentes.

Apache Kafka ACLs tiene el formato «La P principal es la operación O [permitida/denegada] del host H en cualquier recurso R que coincida con el RP». ResourcePattern Si RP no coincide con un recurso R específico, R no tiene asociado y ACLs, por lo tanto, solo los superusuarios pueden acceder a R. Para cambiar este comportamiento de Apache Kafka, establezca la propiedad en `allow.everyone.if.no.acl.found` true. Amazon MSK lo establece como true de forma predeterminada. Esto significa que con los clústeres de Amazon MSK, si no se establece ACLs un recurso de forma explícita, todos los principales pueden acceder a este recurso. Si habilita ACLs un recurso, solo los principales autorizados pueden acceder a él. Si desea restringir el acceso a un tema y autorizar a un cliente mediante la autenticación mutua TLS, agréguelo ACLs mediante la CLI del autorizador Apache Kafka. Para obtener más información sobre cómo añadir, eliminar y enumerar ACLs, consulte la interfaz de línea de comandos de autorización de Kafka.

Como Amazon MSK configura a los corredores como superusuarios, estos pueden acceder a todos los temas. Esto ayuda a los intermediarios a replicar los mensajes de la partición principal, independientemente de que la `allow.everyone.if.no.acl.found` propiedad esté definida o no para la configuración del clúster.

Adición o eliminación del acceso de lectura y escritura a un tema

1. Añada a sus corredores a la tabla de ACL para que puedan leer todos los temas existentes. ACLs Para conceder a los agentes acceso de lectura a un tema, ejecute el siguiente comando en un equipo cliente que pueda comunicarse con el clúster de MSK.

   Reemplácelo *`Distinguished-Name`* por el DNS de cualquiera de los agentes de arranque del clúster y, a continuación, sustituya la cadena situada antes del primer punto de este nombre distintivo por un asterisco ()*. Por ejemplo, si uno de los agentes de arranque del clúster tiene el DNSb-6.mytestcluster.67281x.c4.kafka.us-east-1.amazonaws.com, sustituya el siguiente *`Distinguished-Name`* comando por. `*.mytestcluster.67281x.c4.kafka.us-east-1.amazonaws.com` Para obtener información sobre cómo obtener los agentes de arranque, consulte the section called “Obtenga los corredores de bootstrap”.

   ```
   <path-to-your-kafka-installation>/bin/kafka-acls.sh --bootstrap-server
    BootstrapServerString --add --allow-principal "User:CN=Distinguished-Name" --
   operation Read --group=* --topic Topic-Name
   ```

2. Para conceder a una aplicación cliente acceso de lectura a un tema, ejecute el siguiente comando en el equipo cliente. Si utiliza la autenticación TLS mutua, utilice la misma *Distinguished-Name* que utilizó al crear la clave privada.

```
<path-to-your-kafka-installation>/bin/kafka-acls.sh --bootstrap-server
 BootstrapServerString --add --allow-principal "User:CN=Distinguished-Name" --
operation Read --group=* --topic Topic-Name
```

   Para eliminar el acceso de lectura, puede ejecutar el mismo comando, sustituyendo `--add` por `--remove`.

3. Para conceder acceso de escritura a un tema, ejecute el siguiente comando en su máquina del cliente. Si utilizas la autenticación TLS mutua, usa la misma *Distinguished-Name* que usaste cuando creaste la clave privada.

```
<path-to-your-kafka-installation>/bin/kafka-acls.sh --bootstrap-server
 BootstrapServerString --add --allow-principal "User:CN=Distinguished-Name" --
operation Write --topic Topic-Name
```

   Para eliminar el acceso de escritura, puede ejecutar el mismo comando, sustituyendo `--add` por `--remove`.

Modificación del grupo de seguridad de un clúster de Amazon MSK

En esta página, se explica cómo cambiar el grupo de seguridad de un clúster de MSK existente. Es posible que tenga que cambiar el grupo de seguridad de un clúster para proporcionar acceso a un determinado conjunto de usuarios o para limitar el acceso al clúster. Para obtener más información sobre los grupos de seguridad, consulte Grupos de seguridad de la VPC en la Guía de usuario de Amazon VPC.

1. Utilice la ListNodesAPI o el comando list-nodes del AWS CLI para obtener una lista de los corredores de su clúster. Los resultados de esta operación incluyen las interfaces IDs de red elásticas (ENIs) asociadas a los corredores.
2. Inicia sesión en la EC2 consola de Amazon AWS Management Console y ábrela en https://console.aws.amazon.com/ec2/.
3. En la lista desplegable situada cerca de la esquina superior derecha de la pantalla, seleccione la región en la que se implementa el clúster.
4. En el panel izquierdo, en Red y seguridad, seleccione Interfaces de red.

5. Seleccione la primera ENI que obtuvo en el primer paso. Seleccione el menú Acciones en la parte superior de la pantalla y, a continuación, seleccione Cambiar grupos de seguridad. Asigne el nuevo grupo de seguridad a esta ENI. Repite este paso para cada uno de los ENIs que obtuviste en el primer paso.

> **Note**
>
> Los cambios que realices en el grupo de seguridad de un clúster mediante la EC2 consola de Amazon no se reflejan en la consola de MSK, en la sección Configuración de red.

6. Configure las reglas del nuevo grupo de seguridad para garantizar que sus clientes tengan acceso a los agentes. Para obtener información acerca de cómo establecer reglas de grupo de seguridad, consulte Adición, eliminación y actualización de reglas en la Guía del usuario de Amazon VPC.

> **Important**
>
> Si cambia el grupo de seguridad asociado a los agentes de un clúster y, a continuación, agrega nuevos agentes a ese clúster, Amazon MSK asocia los nuevos agentes al grupo de seguridad original que estaba asociado al clúster cuando se creó el clúster. Sin embargo, para que un clúster funcione correctamente, todos sus agentes deben estar asociados al mismo grupo de seguridad. Por lo tanto, si añade nuevos agentes después de cambiar el grupo de seguridad, debe volver a seguir el procedimiento anterior y actualizar ENIs los agentes nuevos.

Controle el acceso a ZooKeeper los nodos de Apache en su clúster de Amazon MSK

Por motivos de seguridad, puede limitar el acceso a los ZooKeeper nodos de Apache que forman parte de su clúster de Amazon MSK. Para limitar el acceso a los nodos, puede asignarles un grupo de seguridad independiente. A continuación, puede decidir quién obtiene acceso a ese grupo de seguridad.

> **Important**
>
> Esta sección no se aplica a los clústeres que se ejecutan en KRaft modo. Consulte the section called “KRaft modo ”.

Este tema contiene las siguientes secciones:

- Para colocar ZooKeeper los nodos de Apache en un grupo de seguridad independiente
- Uso de la seguridad TLS con Apache ZooKeeper

Para colocar ZooKeeper los nodos de Apache en un grupo de seguridad independiente

Para limitar el acceso a ZooKeeper los nodos de Apache, puede asignarles un grupo de seguridad independiente. Puede establecer reglas del grupo de seguridad para elegir quién tiene acceso a este nuevo grupo de seguridad.

1. Obtenga la cadena de ZooKeeper conexión de Apache para su clúster. Para aprender a hacerlo, consulte the section called “ZooKeeper modo”. La cadena de conexión contiene los nombres DNS de sus ZooKeeper nodos de Apache.
2. Utilice una herramienta como `host` o `ping` para convertir los nombres de los DNS que obtuvo en el paso anterior a las direcciones IP. Guarde estas direcciones IP porque las necesitará más adelante en este procedimiento.
3. Inicia sesión en la EC2 consola de Amazon AWS Management Console y ábrela en https://console.aws.amazon.com/ec2/.
4. En el panel izquierdo, en NETWORK & SECURITY (Red y seguridad), seleccione Network Interfaces (Interfaces de red).
5. En el campo de búsqueda situado encima de la tabla de interfaces de red, escriba el nombre del clúster y, a continuación, escriba return. Esto limita el número de interfaces de red que aparecen en la tabla a las interfaces asociadas al clúster.
6. Marque la casilla de verificación al principio de la fila correspondiente a la primera interfaz de red de la lista.
7. En el panel de detalles de la parte inferior de la página, busca la IPv4 IP privada principal. Si esta dirección IP coincide con una de las direcciones IP que obtuvo en el primer paso de este procedimiento, significa que esta interfaz de red está asignada a un ZooKeeper nodo de Apache que forma parte de su clúster. De lo contrario, anule la selección de la casilla de verificación

situada junto a esta interfaz de red y seleccione la siguiente interfaz de red de la lista. El orden en el que selecciona las interfaces de red no importa. En los siguientes pasos, realizará las mismas operaciones en todas las interfaces de red que estén asignadas a ZooKeeper los nodos de Apache, una por una.

8. Cuando seleccione una interfaz de red que corresponda a un ZooKeeper nodo de Apache, elija el menú Acciones en la parte superior de la página y, a continuación, elija Cambiar grupos de seguridad. Asigne un nuevo grupo de seguridad a esta interfaz de red. Para obtener más información acerca de la creación de grupos de seguridad, consulte Creación de un grupo de seguridad en la documentación de Amazon VPC.
9. Repita el paso anterior para asignar el mismo grupo de seguridad nuevo a todas las interfaces de red asociadas a ZooKeeper los nodos Apache del clúster.
10. Ahora puede elegir quién tiene acceso a este nuevo grupo de seguridad. Para obtener información acerca de cómo establecer reglas de grupo de seguridad, consulte Adición, eliminación y actualización de reglas en la documentación de Amazon VPC.

Uso de la seguridad TLS con Apache ZooKeeper

Puede utilizar la seguridad TLS para el cifrado en tránsito entre sus clientes y sus nodos de Apache ZooKeeper . Para implementar la seguridad TLS en sus ZooKeeper nodos de Apache, haga lo siguiente:

- Los clústeres deben usar la versión 2.5.1 o posterior de Apache Kafka para usar la seguridad TLS con Apache. ZooKeeper
- Active la seguridad TLS al crear o configurar el clúster. Los clústeres creados con la versión 2.5.1 o posterior de Apache Kafka con TLS activado utilizan automáticamente la seguridad TLS con los puntos de conexión de Apache. ZooKeeper Para obtener información sobre la configuración de la seguridad TLS, consulte Introducción al cifrado de Amazon MSK.
- Recupere los puntos finales TLS de Apache mediante la operación. ZooKeeper DescribeCluster
- Cree un archivo ZooKeeper de configuración de Apache para usarlo con las `kafka-acls.sh`herramientas `kafka-configs.sh` y o con el ZooKeeper shell. Con cada herramienta, se utiliza el `--zk-tls-config-file` parámetro para especificar la ZooKeeper configuración de Apache.

  El siguiente ejemplo muestra un archivo de ZooKeeper configuración típico de Apache:

```
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
```

- Para otros comandos (como`kafka-topics`), debe usar la variable de `KAFKA_OPTS` entorno para configurar ZooKeeper los parámetros de Apache. El siguiente ejemplo muestra cómo configurar la variable de `KAFKA_OPTS` entorno para pasar ZooKeeper los parámetros de Apache a otros comandos:

  ```
  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"
  ```

  Después de configurar la variable de entorno `KAFKA_OPTS`, puede utilizar los comandos de la CLI con normalidad. En el siguiente ejemplo, se crea un tema de Apache Kafka con la ZooKeeper configuración de Apache de la variable de `KAFKA_OPTS` entorno:

  ```
  <path-to-your-kafka-installation>/bin/kafka-topics.sh --create --
  zookeeper ZooKeeperTLSConnectString --replication-factor 3 --partitions 1 --topic
   AWSKafkaTutorialTopic
  ```

> **Note**
>
> Los nombres de los parámetros que utiliza en el archivo de ZooKeeper configuración de Apache y los que utiliza en la variable de `KAFKA_OPTS` entorno no son coherentes. Preste atención a los nombres que utiliza con los parámetros del archivo de configuración y la variable de entorno `KAFKA_OPTS`.

Para obtener más información sobre cómo acceder a ZooKeeper los nodos de Apache con TLS, consulte el artículo KIP-515: Habilitar el cliente ZK para utilizar la nueva autenticación compatible con TLS.

Validación de la conformidad de Amazon Managed Streaming para Apache Kafka

Los auditores externos evalúan la seguridad y la conformidad de Amazon Managed Streaming para Apache Kafka en distintos programas de conformidad de AWS . Estos incluyen PCI y HIPAA BAA.

Para obtener una lista de AWS los servicios incluidos en el ámbito de los programas de conformidad específicos, consulte Amazon Services in Scope by Compliance Program . Para obtener información general, consulte Programas de AWS cumplimiento > Programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Su responsabilidad de conformidad al utilizar Amazon MSK viene determinada por la confidencialidad de sus datos, los objetivos de conformidad de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Security and Compliance Quick Start Guides (Guías de inicio rápido de seguridad y conformidad) (Guías de inicio rápido de seguridad y conformidad): Estas guías de implementación analizan las consideraciones en materia de arquitectura y proporcionan los pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- Documento técnico sobre cómo diseñar una arquitectura basada en la seguridad y el cumplimiento de la HIPAA: en este documento técnico se describe cómo pueden utilizar las empresas para crear aplicaciones que cumplan con la HIPAA. AWS
- AWS Recursos de cumplimiento Recursos de de trabajo y guías puede aplicarse a su sector y ubicación.
- Evaluación de los recursos con las reglas de la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub— Este AWS servicio proporciona una visión integral del estado de su seguridad AWS que le ayuda a comprobar su conformidad con los estándares y las mejores prácticas del sector de la seguridad.

Resiliencia en Amazon Managed Streaming para Apache Kafka

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. AWS Las regiones proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación

por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte Infraestructura global.AWS

Seguridad de la infraestructura en Amazon Managed Streaming para Apache Kafka

Como servicio gestionado, Amazon Managed Streaming for Apache Kafka Kafka está protegido por los procedimientos de seguridad de red AWS global que se describen en el documento técnico Amazon Web Services: Overview of Security Processes.

Utiliza las llamadas a la API AWS publicadas para acceder a Amazon MSK a través de la red. Los clientes deben ser compatibles con la seguridad de la capa de transporte (TLS) 1.0 o una versión posterior. Recomendamos TLS 1.2 o una versión posterior. Los clientes también deben ser compatibles con conjuntos de cifrado con confidencialidad directa total (PFS) tales como Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puedes utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Configuración aprovisionada de Amazon MSK

Amazon MSK proporciona configuraciones predeterminadas para los corredores, los temas y los nodos de metadatos. También puede crear configuraciones personalizadas y utilizarlas para crear nuevos clústeres de MSK o para actualizar clústeres existentes. Una configuración de MSK está compuesta por un conjunto de propiedades y sus valores correspondientes. Según el tipo de agente que utilice en su clúster, hay un conjunto diferente de valores de configuración predeterminados y un conjunto diferente de configuraciones que puede modificar. Consulte las secciones siguientes para obtener más información sobre cómo configurar sus corredores Standard y Express.

Temas

- Configuraciones de bróker estándar
- Configuraciones de Express Broker
- Operaciones de configuración del bróker

Configuraciones de bróker estándar

En esta sección se describen las propiedades de configuración de los corredores estándar.

Temas

- Configuraciones personalizadas de Amazon MSK
- Configuración predeterminada de Amazon MSK
- Directrices para la configuración a nivel de tema del almacenamiento en niveles de Amazon MSK

Configuraciones personalizadas de Amazon MSK

Amazon MSK le permite crear una configuración de MSK personalizada en la que se establecen las siguientes propiedades. Las propiedades que no se establecen de forma explícita obtienen los valores que tienen en the section called “Configuración predeterminada de Amazon MSK”. Para obtener más información acerca de las propiedades de configuración, consulte Configuración de Apache Kafka.

Propiedades de configuración de Apache Kafka que se pueden establecer

Nombre	Descripción
allow.everyone.if.no.acl.found	Si desea establecer esta propiedad en `false`, primero asegúrese de definir Apache Kafka ACLs para su clúster. Si establece esta propiedad en Apache Kafka `false` y no lo define primero ACLs, perderá el acceso al clúster. Si eso ocurre, puede volver a actualizar la configuración y establecer esta propiedad en `true` para recuperar el acceso al clúster.
auto.create.topics.enable	Habilita la creación automática de temas en el servidor.
compression.type	El tipo de compresión final para un tema determinado. Puede establecer esta propiedad en los códecs de compresión estándar (`gzip`, `snappy`, `lz4` y `zstd`). Además, acepta `uncompressed` . Este valor equivale a la

Nombre	Descripción
	ausencia de compresión. Si establece el valor en `producer`, se retendrá el códec de compresión original que configuró el productor.
connections.max.idle.ms	Tiempo de espera de las conexiones inactivas en milisegundos. Los subprocesos del procesador del socket del servidor cierran las conexiones que están inactivas durante un tiempo superior al valor establecido para esta propiedad.
default.replication.factor	El factor de replicación predeterminado de los temas que se han creado automáticamente.
delete.topic.enable	Habilita la operación para eliminar un tema. Si desactiva esta configuración, no podrá eliminar un tema usando la herramienta de administr ación.
group.initial.rebalance.delay.ms	Cantidad de tiempo que el coordinador del grupo espera para que otros consumidores de datos se unan a un nuevo grupo antes de que el coordinador del grupo lleve a cabo el primer reequilibrio. Un retraso superior implica posiblemente menos reequilibrios, pero aumenta el tiempo hasta que el procesamiento comience.
group.max.session.timeout.ms	Tiempo de espera máximo de la sesión para los consumidores registrados. Unos tiempos de espera superiores proporcionan a los consumidores más tiempo para procesar los mensajes entre latidos, pero se requiere más tiempo para detectar errores.

Nombre	Descripción
group.min.session.timeout.ms	Tiempo de espera mínimo de la sesión para los consumidores registrados. Unos tiempos de espera inferiores se traducen en una detección de errores más rápida, pero se requieren latidos de consumidores más frecuentes. Esto puede agotar los recursos del agente.
leader.imbalance.per.broker.percentage	La proporción de desequilibrio del líder permitida por agente. El controlador desencade na un equilibrio del líder si supera este valor por agente. Este valor se especifica en porcentaje.
log.cleaner.delete.retention.ms	Cantidad de tiempo que desea que Apache Kafka conserve los registros eliminados. El valor mínimo es 0.
log.cleaner.min.cleanable.ratio	Esta propiedad de configuración puede tener valores entre 0 y 1. Este valor determina la frecuencia con la que el compactador de registros intenta limpiar el registro (si la compactación de registros está habilitada). De forma predeterminada, Apache Kafka evita limpiar un registro si se ha compactado más del 50 % de este. Esta proporción limita el espacio máximo que el registro desperdicia con duplicados (al 50 %, esto significa que como máximo el 50 % del registro podría estar duplicado). Una proporción mayor se traduce en limpiezas más eficaces y menos frecuente s, pero también implica un gasto de espacio superior en el registro.

Nombre	Descripción
log.cleanup.policy	La política de limpieza predeterminada de los segmentos que superan el periodo de retención . Una lista de políticas válidas separadas por comas. Las políticas válidas son `delete` y `compact`. En el caso de los clústeres habilitados para el almacenamiento por niveles, la única política válida es `delete`.
log.flush.interval.messages	Número de mensajes acumulados en una partición de registro antes de que se vacíen en el disco.
log.flush.interval.ms	Tiempo máximo en milisegundos que un mensaje de cualquier tema se conserva en la memoria antes de vaciarlo en el disco. Si no establece este valor, se utiliza el valor de log.flush.scheduler.interval.ms. El valor mínimo es 0.
log.message.timestamp.difference.max.ms	Esta configuración está obsoleta en Kafka 3.6.0. Se han agregado dos configuraciones, `log.message.timestamp.before.max.ms` y `log.message.timestamp.after.max.ms`. La diferencia temporal máxima entre la marca temporal que se produce cuando un agente recibe un mensaje y la marca temporal que se especifica en el mensaje. Si es log.message.timestamp.type=CreateTime, se rechaza un mensaje si la diferencia en la marca temporal supera este umbral. Esta configuración se ignora si es LogAppendTime log.message.timestamp.type=.

Nombre	Descripción
log.message.timestamp.type	Especifica si la marca temporal del mensaje es la hora de creación del mensaje o la hora de adición del registro. Los valores permitidos son `CreateTime` y `LogAppendTime` .
log.retention.bytes	Tamaño máximo del registro antes de eliminarlo.
log.retention.hours	Número de horas que se conserva un archivo de registro antes de eliminarlo, cantidad terciaria de la propiedad log.retention.ms.
log.retention.minutes	Número de minutos que se conserva un archivo de registro antes de eliminarlo, cantidad secundaria de la propiedad log.retention.ms. Si no establece este valor, se utiliza el valor de log.retention.hours.
log.retention.ms	Número de milisegundos que se conserva un archivo de registro antes de eliminarlo (en milisegundos). Si no se establece, se utiliza el valor de log.retention.minutes.
log.roll.ms	Tiempo máximo antes de que un segmento de registro nuevo se implemente (en milisegundos). Si no establece esta propiedad, se utiliza el valor de log.roll.hours. El valor mínimo posible de esta propiedad es 1.
log.segment.bytes	Tamaño máximo de un único archivo de registro.
max.incremental.fetch.session.cache.slots	Número máximo de sesiones de recuperación incrementales que se conservan.

Nombre	Descripción
message.max.bytes	Tamaño de lote de registros más grande que admite Kafka. Si aumenta este valor y hay consumidores anteriores a 0.10.2, también debe aumentar el tamaño de recuperación de los consumidores para que se puedan recuperar lotes de registros de este tamaño. La versión de formato de mensaje más reciente siempre agrupa los mensajes en lotes para aumentar la eficacia. Las versiones de formato de mensaje anteriores no agrupan los registros sin comprimir en lotes y, en este caso, este límite se aplica únicamente a un solo registro. Puede establecer este valor por tema con la configuración de temas max.message.bytes.
min.insync.replicas	Cuando un productor establece las confirmac iones en `"all"` (o `"-1"`), el valor de min.insyn c.replicas especifica el número mínimo de réplicas que debe confirmar una escritura para que se considere como correcta. Si no se puede alcanzar este mínimo, el productor hace una excepción (o bien). NotEnoughReplicas NotEnoughReplicasAfterAppend Puede utilizar valores de min.insync.replicas y confirmaciones para reforzar las garantías de durabilidad. Por ejemplo, puede crear un tema con un factor de replicación de 3, establecer min.insync.replicas en 2 y producirlo con las confirmaciones de `"all"`. Esto garantiza que el productor emita una excepción si la mayoría de las réplicas no reciben una escritura.

Nombre	Descripción
num.io.threads	El número de subprocesos que utiliza el servidor para procesar las solicitudes, puede incluir la E/S del disco.
num.network.threads	El número de subprocesos que utiliza el servidor para recibir solicitudes desde la red y enviarle las respuestas a dichas solicitudes.
num.partitions	Número predeterminado de particiones de registro por tema.
num.recovery.threads.per.data.dir	El número de subprocesos por directorio de datos que se va a utilizar para la recuperación de registros en el arranque y para el vaciado en el apagado.
num.replica.fetchers	El número de subprocesos del recuperador que se utilizan para replicar los mensajes desde un agente de origen. Si aumentas este valor, puedes aumentar el grado de I/O paralelismo en el bróker seguidor.
offsets.retention.minutes	Después de que un grupo de consumidores pierda todos sus consumidores (es decir, se quede vacío), sus compensaciones se conservan durante este periodo de retención antes de que se descarten. Para los consumido res independientes (es decir, los que utilizan asignación manual), las compensaciones vencen después de la hora de la última confirmación más este periodo de retención.

Nombre	Descripción
offsets.topic.replication.factor	El factor de replicación del tema de compensación. Establezca este valor en un valor más alto para garantizar la disponibilidad. Se produce un error en la creación del tema interno hasta que el tamaño del clúster cumpla este requisito de factor de replicación.
replica.fetch.max.bytes	Número de bytes de los mensajes para intentar recuperar cada partición. Esto no es un valor máximo absoluto. Si el primer lote de registros de la primera partición que no está vacía de la recuperación es superior a este valor, el lote de registros se devuelve para asegurar el progreso. Las propiedades message.max.bytes (configuración del agente) o max.messa ge.bytes (configuración del tema) definen el tamaño de lote de registro máximo que acepta el agente.
replica.fetch.response.max.bytes	El número máximo de bytes previsto para la respuesta de recuperación completa. Los registros se recuperan en lotes y, si el primer lote de registro de la primera partición que no está vacía de la recuperación es superior a este valor, el lote de registro se devolverá para asegurar que se lleva a cabo el progreso. Esto no es un valor máximo absoluto. Las propiedad es message.max.bytes (configuración del agente) o max.message.bytes (configuración del tema) especifican el tamaño de lote de registro máximo que acepta el agente.

Nombre	Descripción
replica.lag.time.max.ms	Si un seguidor no ha enviado ninguna solicitud de recuperación o no ha consumido hasta la compensación final del registro del líder durante al menos este número de milisegun dos, el líder elimina el seguidor del ISR. MinValue: 10000 MaxValue = 30000
replica.selector.class	El nombre de clase totalmente cualificado que se implementa. ReplicaSelector El agente utiliza este valor para encontrar la réplica de lectura preferida. Si utiliza la versión 2.4.1 o superior de Apache Kafka y desea permitir que los consumidores puedan recuperar desde la réplica más cercana, establezca esta propiedad en `org.apache.kafka.common.replica.RackAwareReplicaSelector` . Para obtener más información, consulte the section called “Versión 2.4.1 de Apache Kafka (utilice 2.4.1.1 en su lugar)”.
replica.socket.receive.buffer.bytes	El búfer de recepción de sockets para las solicitudes de red.
socket.receive.buffer.bytes	Búfer SO_RCVBUF de los sockets del servidor de sockets. El valor mínimo que puede establecer para esta propiedad es -1. Si el valor es -1, Amazon MSK usa el sistema operativo predeterminado.
socket.request.max.bytes	El número máximo de bytes de una solicitud de conector.

Nombre	Descripción
socket.send.buffer.bytes	Búfer SO_SNDBUF de los sockets del servidor de sockets. El valor mínimo que puede establecer para esta propiedad es -1. Si el valor es -1, Amazon MSK usa el sistema operativo predeterminado.
transaction.max.timeout.ms	Tiempo de espera máximo para las transacci ones. Si el tiempo de transacción solicitado por un cliente supera este valor, el bróker devuelve un error. InitProducerIdRequest Esto evita que un cliente experimente un tiempo de espera demasiado grande, lo que puede detener la lectura de los temas que se incluyen en la transacción por parte de los consumidores.
transaction.state.log.min.isr	Se ha anulado la configuración min.insyn c.replicas para el tema de la transacción.
transaction.state.log.replication.factor	El factor de replicación del tema de transacci ón. Establezca esta propiedad en un valor más elevado para aumentar la disponibilidad. Se produce un error en la creación del tema interno hasta que el tamaño del clúster cumpla este requisito de factor de replicación.

Nombre	Descripción
transactional.id.expiration.ms	El tiempo en milisegundos que el coordinador de transacciones espera para recibir cualquier actualización del estado de la transacción actual antes de que venza el ID de la transacci ón del coordinador. Esta configuración también influye en la caducidad del ID de productor , ya que hace que el productor IDs caduque cuando este tiempo transcurre después de la última escritura con el ID de productor indicado. IDs Es posible que el productor caduque antes si se elimina la última escritura del ID de productor debido a la configuración de retención del tema. El valor mínimo de esta propiedad es de 1 milisegundo.
unclean.leader.election.enable	Indica si las réplicas que no están incluidas en el conjunto de ISR deben servir de líder como último recurso, aunque esto pueda provocar la pérdida de datos.
zookeeper.connection.timeout.ms	ZooKeeper clústeres de modos. Tiempo máximo que el cliente espera para establecer una conexión. ZooKeeper Si no establece este valor, se utiliza el valor de zookeeper.session. timeout.ms. MinValue = 6000 MaxValue (incluido) = 18000 Se recomienda establecer este valor en 10.000 en T3.small para evitar el tiempo de inactividad del clúster.

Nombre	Descripción
zookeeper.session.timeout.ms	ZooKeeper clústeres de modos. El tiempo de espera ZooKeeper de la sesión de Apache en milisegundos. MinValue = 6000 MaxValue (incluido) = 18000

Para obtener información acerca de cómo puede crear una configuración de MSK personalizada, enumerar todas las configuraciones o describirlas, consulte the section called “Operaciones de configuración de corredores”. Para crear un clúster de MSK con una configuración personalizada de MSK o para actualizar un clúster con una nueva configuración personalizada, consulte the section called “Características y conceptos clave”.

Cuando actualiza su clúster de MSK existente con una configuración personalizada de MSK, Amazon MSK se restablece cuando es necesario y utiliza las prácticas recomendadas para reducir el tiempo de inactividad del cliente. Por ejemplo, después de que Amazon MSK restablezca cada agente, el servicio intenta dejar que el agente se ponga al día con los datos que ha podido perder durante la actualización de la configuración antes de pasar al siguiente agente.

Configuración dinámica de Amazon MSK

Además de las propiedades de configuración que ofrece Amazon MSK, también puede establecer propiedades de configuración del agente y del clúster de forma dinámica que no requieran un restablecimiento del agente. Puede establecer de forma dinámica algunas propiedades de configuración. Estas son las propiedades que no están marcadas como de solo lectura en la tabla en Broker Configs en la documentación de Apache Kafka. Para obtener más información acerca de configuraciones dinámicas y ejemplos de comandos, consulte Updating Broker Configs en la documentación de Apache Kafka.

> **Note**
>
> Puede establecer la propiedad `advertised.listeners`, pero no la propiedad `listeners`.

Configuración a nivel de tema de Amazon MSK

Puede utilizar los comandos de Apache Kafka para establecer o modificar propiedades de configuración de nivel de tema para temas nuevos y existentes. Para obtener más información acerca de las propiedades de configuración de temas y ejemplos de cómo establecerlas, consulte Topic-Level Configs en la documentación de Apache Kafka.

Configuración predeterminada de Amazon MSK

Al crear un clúster de MSK sin especificar una configuración de MSK personalizada, Amazon MSK crea y utiliza una configuración predeterminada con los valores que se muestran en la tabla siguiente. Con respecto a las propiedades que no están en esta tabla, Amazon MSK utiliza los valores predeterminados asociados a su versión de Apache Kafka. Para obtener una lista de estos valores predeterminados, consulte Configuración de Apache Kafka.

Valores de configuración predeterminados

Nombre	Descripción	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
allow.everyone.if.no.acl.found	Si ningún patrón de recursos coincide con un recurso específico, el recurso no tiene ninguna asociación. ACLs En este caso, si establece esta propiedad como `true`, todos los usuarios pueden acceder al recurso, no solo los superusuarios.	`true`	`true`

Nombre	Descripción	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
auto.create.topics.enable	Habilita la creación automática de un tema en el servidor.	`false`	`false`
auto.leader.rebalance.enable	Habilita el equilibrio automático del líder. Un subproceso de fondo comprueba e inicia el equilibrio del líder a intervalos regulares, si es necesario.	`true`	`true`
default.replication.factor	Factores de replicación predeterminados de los temas que se han creado automáticamente.	3 para clústeres en 3 zonas de disponibilidad y 2 para clústeres en 2 zonas de disponibilidad.	3 para clústeres en 3 zonas de disponibilidad y 2 para clústeres en 2 zonas de disponibilidad.

Nombre	Descripción	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
local.retention.bytes	El tamaño máximo de los segmentos de registro locales de una partición antes de que elimine los segmentos antiguos. Si no establece este valor, se utiliza el valor de log.retention.bytes. El valor efectivo debe ser siempre menor o igual que el valor de log.retention.bytes. Un valor predeterminado de -2 indica que no hay límite de retención local. Esto corresponde a la configuración retention.ms/bytes de -1. Las propiedades local.retention.ms y local.retention.bytes son similares a las de log.retention, ya que se utilizan para determinar cuánto tiempo deben permanecer los	-2 para un número ilimitado	-2 para un número ilimitado

Nombre	Descripción	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
	segmentos de registro en el almacenamiento local. Las configuraciones de log.retention.* existentes son configuraciones de retención para la partición de temas. Esto incluye el almacenamiento local y remoto. Valores válidos: números enteros en [-2; +Inf]		

Nombre	Descripción	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
local.retention.ms	El número de milisegundos para retener el segmento de registro locale antes de la eliminación. Si no establece este valor, Amazon MSK utilizará el valor de log.retention.ms. El valor efectivo debe ser siempre menor o igual que el valor de log.retention.bytes. Un valor predeterminado de -2 indica que no hay límite de retención local. Esto corresponde a la configuración retention.ms/bytes de -1. Los valores de local.retention.ms y local.retention.bytes son similares a los de log.retention. MSK usa esta configuración para determinar cuánto tiempo deben	-2 para un número ilimitado	-2 para un número ilimitado

Nombre	Descripción	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
	permanecer los segmentos de registro en el almacenamiento local. Las configuraciones de log.retention.* existentes son configuraciones de retención para la partición de temas. Esto incluye el almacenamiento local y remoto. Los valores válidos son números enteros mayores que 0.		

Nombre	Descripción	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
log.message.timestamp.difference.max.ms	Esta configuración está obsoleta en Kafka 3.6.0. Se han agregado dos configuraciones, `log.message.timestamp.before.max.ms` y `log.message.timestamp.after.max.ms` ,. Diferencia máxima permitida entre la marca temporal que se produce cuando un agente recibe un mensaje y la marca temporal que se especifica en el mensaje. Si es log.message.timestamp.type=CreateTime, se rechazará un mensaje si la diferencia en la marca temporal supera este umbral. Esta configuración se	9223372036854775807	86400000 para Kafka con niveles 2.8.2 y Kafka 3.7.x con niveles.

Nombre	Descripción	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
	ignora si es log.message.timestamp.type=. LogAppendTime La diferencia de marca temporal máxima permitida no debe ser superior a la de log.retention.ms para evitar la acumulación innecesariamente frecuente de registros.		
log.segment.bytes	El tamaño máximo de un único archivo de registro.	1073741824	134217728

Nombre	Descripción	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
min.insync.replicas	Cuando un productor establece el valor de las confirmaciones (la confirmación que el productor recibe del agente de Kafka) en `"all"` (o `"-1"`), el valor de min.insync.replicas especifica el número mínimo de réplicas que debe confirmar una escritura para que se considere como correcta. Si este valor no cumple con este mínimo, el productor hace una excepción (una de las dos NotEnough Replicas opciones). NotEnoughReplicasAfterAppend Cuando utiliza de forma conjunta los valores de min.insync.replicas y las confirmaciones, puede aplicar unas garantías de durabilid	2 para clústeres en 3 zonas de disponibilidad y 1 para clústeres en 2 zonas de disponibilidad.	2 para clústeres en 3 zonas de disponibilidad y 1 para clústeres en 2 zonas de disponibilidad.

Nombre	Descripción	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
	ad mayores. Por ejemplo, puede crear un tema con un factor de replicación de 3, establecer min.insync.replicas en 2 y producirlo con las confirmaciones de `"all"`. Esto garantiza que el productor emita una excepción si la mayoría de las réplicas no reciben una escritura.		
num.io.threads	Número de subprocesos que utiliza el servidor para producir las solicitudes, puede incluir la E/S del disco.	8	max (8, vCPUs), donde v CPUs depende del tamaño de la instancia del bróker
num.network.threads	El número de subprocesos que utiliza el servidor para recibir solicitudes desde la red y enviarle las respuestas.	5	max (5, vCPUs /2) donde v CPUs depende del tamaño de la instancia del bróker

Nombre	Descripción	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
num.partitions	Número predeterminado de particiones de registro por tema.	1	1
num.replica.fetchers	Número de subprocesos de búsqueda utilizados para replicar los mensajes de un agente fuente. Si aumentas este valor, puedes aumentar el grado de I/O paralelismo en el agente seguidor.	2	max (2, vCPUs /4) donde v CPUs depende del tamaño de la instancia del broker
remote.log.msk.disable.policy	Se usa con remote.storage.enable para deshabilitar el almacenamiento por niveles. Establezca esta política en Eliminar para indicar que los datos del almacenamiento por niveles se eliminarán al establecer remote.storage.enable en false.	N/A	Ninguno

Nombre	Descripción	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
remote.log.reader.threads	Tamaño del grupo de subprocesos del lector de registros remoto, que se utiliza para programar tareas a fin de recuperar datos del almacenamiento remoto.	N/A	max (10, v CPUs * 0.67) donde v CPUs depende del tamaño de la instancia del broker

Nombre	Descripción	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
remote.storage.enable	Habilita el almacenamiento por niveles (remoto) para un tema si se establece en true. Deshabilita el almacenamiento por niveles del tema si se establece en false y remote.log.msk.disable.policy se establece en Eliminar. Al deshabilitar el almacenamiento por niveles, se eliminan los datos del almacenamiento remoto. Cuando deshabilita el almacenamiento por niveles para un tema, no podrá volver a habilitarlo.	false	false

Nombre	Descripción	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
replica.lag.time.max.ms	Si un seguidor no ha enviado ninguna solicitud de recuperación o no ha consumido hasta la compensación final del registro del líder durante al menos este número de milisegundos, el líder elimina el seguidor del ISR.	30000	30000

Nombre	Descripción	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
retention.ms	Campo obligatorio. El tiempo mínimo es de 3 días. No hay ningún valor predeterminado porque la configuración es obligatoria. Amazon MSK usa el valor de retention.ms con local.retention.ms para determinar cuándo se transfier en los datos del almacenamiento local al almacenamiento por niveles. El valor de local.retention.ms especifica cuándo se transferir los datos del almacenamiento local al almacenamiento por niveles. El valor de retention.ms especifica cuándo se deben eliminar los datos del almacenamiento en niveles (es decir, eliminarlos del clúster). Valores	Mínimo de 259 200 000 milisegundos (3 días). -1 para una retención infinita.	Mínimo de 259 200 000 milisegundos (3 días). -1 para una retención infinita.

Nombre	Descripción	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
	válidos: números enteros en [-1; +Inf]		
socket.receive.buffer.bytes	El búfer SO_RCVBUF de los sockets del servidor de sockets. Si el valor es -1, se utiliza el sistema operativo predeterminado.	102400	102400
socket.request.max.bytes	Número máximo de bytes de una solicitud de conector.	104857600	104857600
socket.send.buffer.bytes	El búfer SO_SNDBUF de los sockets del servidor de sockets. Si el valor es -1, se utiliza el sistema operativo predeterminado.	102400	102400
unclean.leader.election.enable	Indica si quiere que las réplicas que no están incluidas en el conjunto de ISR sirvan de líder como último recurso, aunque esto pueda provocar la pérdida de datos.	true	false

Nombre	Descripción	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
zookeeper.session.timeout.ms	El tiempo de espera ZooKeeper de la sesión de Apache en milisegundos.	18000	18000
zookeeper.set.acl	El cliente configurado para usar secure ACLs.	false	false

Para obtener más información acerca de cómo especificar los valores de configuración personalizados, consulte the section called “Configuraciones personalizadas de Amazon MSK”.

Directrices para la configuración a nivel de tema del almacenamiento en niveles de Amazon MSK

Las siguientes son las configuraciones y limitaciones predeterminadas al configurar el almacenamiento por niveles de los temas.

- Amazon MSK no admite tamaños de segmento de registro más pequeños para los temas con el almacenamiento por niveles activado. Si desea crear un segmento, hay un tamaño mínimo de segmento de registro de 48 MiB o un tiempo mínimo de rotación del segmento de 10 minutos. Estos valores se asignan a las propiedades segment.bytes y segment.ms.
- El valor de local.retention. ms/bytes can't equal or exceed the retention.ms/bytes. Esta es la configuración de retención del almacenamiento por niveles.
- El valor predeterminado para local.retention. ms/bytes is -2. This means that the retention.ms value is used for local.retention.ms/bytes. En este caso, los datos permanecen tanto en el almacenamiento local como en el almacenamiento por niveles (una copia en cada uno) y vencen juntos. Para esta opción, se conserva una copia de los datos locales en el almacenamiento remoto. En este caso, los datos leídos del tráfico de consumo provienen del almacenamiento local.
- El valor predeterminado de retention.ms es 7 días. No hay un límite de tamaño predeterminado para retention.bytes.
- El valor mínimo de retention.ms/bytes es -1. Esto significa una retención infinita.

- El valor mínimo de local.retention. ms/bytes is -2. This means infinite retention for local storage. It matches with the retention.ms/bytesestablecer como -1.
- La configuración a nivel de tema retention.ms es obligatoria para los temas con el almacenamiento en niveles activado. El valor mínimo de retention.ms es de 3 días.

Para obtener más información sobre las restricciones de almacenamiento por niveles, consulte. Restricciones y limitaciones de almacenamiento por niveles para clústeres de Amazon MSK

Configuraciones de Express Broker

Apache Kafka tiene cientos de configuraciones de broker que puede utilizar para ajustar el rendimiento de su clúster aprovisionado de MSK. Establecer valores erróneos o subóptimos puede afectar a la confiabilidad y el rendimiento del clúster. Express Brokers mejora la disponibilidad y la durabilidad de los clústeres aprovisionados por MSK al establecer valores óptimos para las configuraciones críticas y protegerlas de los errores de configuración habituales. Existen tres categorías de configuraciones basadas en el acceso de lectura y escritura: configuraciones de lectura/escritura (editables), de solo lectura y de no lectura/escritura. Algunas configuraciones siguen utilizando el valor predeterminado de Apache Kafka para la versión de Apache Kafka que ejecuta el clúster. Las marcamos como Apache Kafka Default.

Temas

- Configuraciones de broker MSK Express personalizadas (acceso de lectura/escritura)
- Configuraciones de solo lectura de Express Brokers

Configuraciones de broker MSK Express personalizadas (acceso de lectura/escritura)

Puede actualizar las configuraciones de los read/write corredores mediante la función de configuración de actualizaciones de Amazon MSK o mediante la API de AlterConfig Apache Kafka. Las configuraciones de los corredores de Apache Kafka son estáticas o dinámicas. Las configuraciones estáticas requieren un reinicio del broker para poder aplicar la configuración, mientras que las configuraciones dinámicas no necesitan un reinicio del broker. Para obtener más información sobre las propiedades de configuración y los modos de actualización, consulte Actualización de las configuraciones del broker.

Temas

- Configuraciones estáticas en los corredores de MSK Express
- Configuraciones dinámicas en Express Brokers

- Configuraciones de nivel temático en Express Brokers

Configuraciones estáticas en los corredores de MSK Express

Puede usar Amazon MSK para crear un archivo de configuración de MSK personalizado para establecer las siguientes propiedades estáticas. Amazon MSK establece y administra todas las demás propiedades que usted no establece. Puede crear y actualizar archivos de configuración estáticos desde la consola de MSK o mediante el comando configuration.

Propiedad	Descripción	Valor predeterminado
allow.everyone.if.no.acl.found	Si desea establecer esta propiedad en false, primero asegúrese de definir Apache Kafka ACLs para su clúster. Si establece esta propiedad en false y no define primero Apache Kafka ACLs, perderá el acceso al clúster. Si eso ocurre, puede volver a actualizar la configuración y establecer esta propiedad en true para recuperar el acceso al clúster.	true
auto.create.topics.enable	Habilita la creación automática de un tema en el servidor.	false
compression.type	Especifique el tipo de compresión final para un tema determinado. Esta configuración acepta los códecs de compresión estándar: gzip, snappy, lz4, zstd. Esta configuración también acepta`uncompressed` , lo que equivale a no comprimir	Apache Kafka es el predeterminado

Propiedad	Descripción	Valor predeterminado
	, y`producer`, lo que significa conservar el códec de compresión original estableci do por el fabricante.	
connections.max.idle.ms	Tiempo de espera de las conexiones inactivas en milisegundos. Los subproces os del procesador del socket del servidor cierran las conexiones que están inactivas durante un tiempo superior al valor establecido para esta propiedad.	Apache Kafka predeterminado
delete.topic.enable	Habilita la operación para eliminar un tema. Si desactiva esta configuración, no podrá eliminar un tema usando la herramienta de administr ación.	Apache Kafka predeterminado
group.initial.rebalance.del ay.ms	Cantidad de tiempo que el coordinador del grupo espera para que otros consumido res de datos se unan a un nuevo grupo antes de que el coordinador del grupo lleve a cabo el primer reequilibrio. Un retraso superior implica posiblemente menos reequilib rios, pero aumenta el tiempo hasta que el procesamiento comience.	Apache Kafka predeterminado

Propiedad	Descripción	Valor predeterminado
group.max.session.timeout.ms	Tiempo de espera máximo de la sesión para los consumidores registrados. Unos tiempos de espera superiores proporcionan a los consumidores más tiempo para procesar los mensajes entre latidos, pero se requiere más tiempo para detectar errores.	Apache Kafka predeterminado
leader.imbalance.per.broker.percentage	La proporción de desequilibrio del líder permitida por agente. El controlador desencadena un equilibrio del líder si supera este valor por agente. Este valor se especifica en porcentaje.	Apache Kafka predeterminado
log.cleanup.policy	La política de limpieza predeterminada de los segmentos que superan el periodo de retención. Una lista de políticas válidas separadas por comas. Las políticas válidas son `delete` y `compact`. Para los clústeres habilitados para el almacenamiento por niveles, la política válida es únicamente. `delete`	Apache Kafka es el valor predeterminado

Propiedad	Descripción	Valor predeterminado
log.message.timestamp.after.max.ms	La diferencia de marca de tiempo permitida entre la marca de tiempo del mensaje y la marca de tiempo del corredor. La marca de tiempo del mensaje puede ser posterior o igual a la marca de tiempo del corredor, con la diferencia máxima permitida determinada por el valor establecido en esta configuración. Si `log.message.timestamp.type=CreateTime`, el mensaje se rechazará si la diferencia en las marcas de tiempo supera este umbral especificado. Esta configuración se ignora si. `log.message.timestamp.type=LogAppendTime`	86400000 (24 * 60 * 60 * 1000 ms, es decir, 1 día)

Propiedad	Descripción	Valor predeterminado
log.message.timestamp.before.max.ms	La diferencia de marca de tiempo permitida entre la marca de tiempo del corredor y la marca de tiempo del mensaje. La marca de tiempo del mensaje puede ser anterior o igual a la marca de tiempo del corredor, con la diferencia máxima permitida determinada por el valor establecido en esta configuración. Si `log.message.timestamp.type=CreateTime` , el mensaje se rechazará si la diferencia en las marcas de tiempo supera este umbral especificado. Esta configuración se ignora si. `log.message.timestamp.type=LogAppendTime`	86400000 (24 * 60 * 60 * 1000 ms, es decir, 1 día)
log.message.timestamp.type	Especifica si la marca temporal del mensaje es la hora de creación del mensaje o la hora de adición del registro. Los valores permitidos son `CreateTime` y `LogAppendTime` .	Apache Kafka es el valor predeterminado
log.retention.bytes	Tamaño máximo del registro antes de eliminarlo.	Apache Kafka predeterminado

Propiedad	Descripción	Valor predeterminado
log.retention.ms	Número de milisegundos que se debe conservar un archivo de registro antes de eliminarlo.	Apache Kafka es el predeterminado
max.connections.per.ip	El número máximo de conexiones permitidas desde cada dirección IP. Esto se puede establecer `0` si hay anulaciones configuradas mediante la `max.connections.per.ip.overrides` propiedad. Si se alcanza el límite, se eliminan las conexiones nuevas desde la dirección IP.	Apache Kafka es el valor predeterminado
max.incremental.fetch.session.cache.slots	Número máximo de sesiones de recuperación incrementales que se conservan.	Apache Kafka predeterminado

Propiedad	Descripción	Valor predeterminado
message.max.bytes	Tamaño de lote de registros más grande que admite Kafka. Si aumenta este valor y hay consumidores anteriores a 0.10.2, también debe aumentar el tamaño de recuperación de los consumidores para que se puedan recuperar lotes de registros de este tamaño. La versión de formato de mensaje más reciente siempre agrupa los mensajes en lotes para aumentar la eficacia. Las versiones de formato de mensaje anteriores no agrupan los registros sin comprimir en lotes y, en este caso, este límite se aplica únicamente a un solo registro. Puede establecer este valor por tema con la `max.message.bytes` configuración a nivel de tema.	Apache Kafka es el predeterminado
num.partitions	Número predeterminado de particiones por tema.	1

Propiedad	Descripción	Valor predeterminado
offsets.retention.minutes	Después de que un grupo de consumidores pierda todos sus consumidores (es decir, se quede vacío), sus compensaciones se conservan durante este periodo de retención antes de que se descarten. Para los consumidores independientes (es decir, aquellos que utilizan la asignación manual), las compensaciones caducan después de la última confirmación más este período de retención.	Apache Kafka es el predeterminado
replica.fetch.max.bytes	Número de bytes de los mensajes para intentar recuperar cada partición. Esto no es un valor máximo absoluto. Si el primer lote de registros de la primera partición que no está vacía de la recuperación es superior a este valor, el lote de registros se devuelve para asegurar el progreso. Las propiedades message.max.bytes (configuración del agente) o max.message.bytes (configuración del tema) definen el tamaño de lote de registro máximo que acepta el agente.	Apache Kafka predeterminado

Propiedad	Descripción	Valor predeterminado
replica.selector.class	El nombre de clase completo que se implementa. ReplicaSelector El agente utiliza este valor para encontrar la réplica de lectura preferida. Si quiere permitir a los consumidores realizar búsquedas desde la réplica más cercana, defina esta propiedad en. `org.apache.kafka.common.replica.RackAwareReplicaSelector`	Apache Kafka es el valor predeterminado
socket.receive.buffer.bytes	El búfer SO_RCVBUF de los sockets del servidor de sockets. Si el valor es -1, se utiliza el sistema operativo predeterminado.	102400
socket.request.max.bytes	Número máximo de bytes de una solicitud de conector.	104857600
socket.send.buffer.bytes	El búfer SO_SNDBUF de los sockets del servidor de sockets. Si el valor es -1, se utiliza el sistema operativo predeterminado.	102400

Propiedad	Descripción	Valor predeterminado
transaction.max.timeout.ms	Tiempo de espera máximo para las transacciones. Si el tiempo de transacción solicitado por un cliente supera este valor, el bróker devuelve un error. InitProducerIdRequest Esto evita que un cliente experimente un tiempo de espera demasiado grande, lo que puede detener la lectura de los temas que se incluyen en la transacción por parte de los consumidores.	Apache Kafka es el valor predeterminado

Propiedad	Descripción	Valor predeterminado
transactional.id.expiration.ms	El tiempo en milisegundos que el coordinador de transacciones espera para recibir cualquier actualización del estado de la transacción actual antes de que venza el ID de la transacción del coordinador. Esta configuración también influye en la caducidad del ID de productor, ya que hace IDs que el productor caduque cuando este tiempo transcurre después de la última escritura con el ID de productor indicado. IDs Es posible que el productor caduque antes si se elimina la última escritura del ID de productor debido a la configuración de retención del tema. El valor mínimo de esta propiedad es de 1 milisegundo.	Apache Kafka es el predeterminado

Configuraciones dinámicas en Express Brokers

Puede utilizar la AlterConfig API de Apache Kafka o la herramienta Kafka-configs.sh para editar las siguientes configuraciones dinámicas. Amazon MSK establece y administra todas las demás propiedades que usted no establece. Puede establecer dinámicamente las propiedades de configuración a nivel de clúster y a nivel de broker que no requieren el reinicio del broker.

Propiedad	Descripción	Valor predeterminado
anunciado .oyentes	Los oyentes se publicará n para que los usen los clientes, si son diferentes de la propiedad de configuración. `listeners` En los entornos de IaaS, es posible que sea diferente de la interfaz a la que se vincula el bróker. Si no está estableci do, se utilizará el valor para los oyentes. A diferencia de los oyentes, no es válido anunciar la metadirección 0.0.0.0. Por otra parte`listeners` , esta propiedad también puede tener puertos duplicado	null

Propiedad	Descripción	Valor predeterminado
	s, de modo que se puede configurar un oyente para que anuncie la dirección de otro. Esto puede resultar útil en algunos casos en los que se utilizan balancead ores de carga externos. Esta propiedad se establece a nivel de bróker.	

Propiedad	Descripción	Valor predeterminado
compression.type	El tipo de compresión final para un tema determinado. Puede establecer esta propiedad en los códecs de compresión estándar (`gzip`, `snappy`, `lz4` y `zstd`). Además, acepta `uncompressed` . Este valor equivale a la ausencia de compresión. Si establece el valor en `producer`, se retendrá el códec de compresión original que configuró el productor.	Apache Kafka es el valor predeterminado

Propiedad	Descripción	Valor predeterminado
log.cleaner.delete.retention.ms	El tiempo que se debe conservar los marcadores de lápidas borrados para temas compactados en registros. Esta configuración también establece un límite en el tiempo que un consumidor debe completar una lectura si comienza con un offset de 0 para garantizar que obtiene una instantánea válida de la etapa final. De lo contrario, es posible que se recopilen lápidas eliminadas antes de que terminen de escanearlas.	86400000 (24 * 60 * 60 * 1000 ms, es decir, 1 día), Apache Kafka Default

Propiedad	Descripción	Valor predeterminado
log.cleaner.min.compaction.lag.ms	El tiempo mínimo que un mensaje permanecerá sin compactar en el registro. Esta configuración solo se aplica a los registros que se están compactando.	0, Apache Kafka es el valor predeterminado
log.cleaner.max.compaction.lag.ms	El tiempo máximo que un mensaje permanecerá inapto para compactarse en el registro. Esta configuración solo se aplica a los registros que se están compactando. Esta configuración estaría limitada en el rango de [7 días, Long.Max].	9223372036854775807, Apache Kafka es el valor predeterminado

Propiedad	Descripción	Valor predeterminado
log.clean up.policy	La política de limpieza predeterm inada de los segmentos que superan el periodo de retención . Una lista de políticas válidas separadas por comas. Las políticas válidas son `delete` y `compact`. Para los clústeres habilitad os para almacenam iento por niveles, la política válida es únicamente. `delete`	Apache Kafka es el valor predeterminado

Propiedad	Descripción	Valor predeterminado
log.message.timestamp.after.max.ms	La diferencia de marca de tiempo permitida entre la marca de tiempo del mensaje y la marca de tiempo del corredor. La marca de tiempo del mensaje puede ser posterior o igual a la marca de tiempo del corredor, con la diferencia máxima permitida determinada por el valor establecido en esta configuración. Si `log.message.timestamp.type=CreateTime`, el mensaje se rechazará si la diferencia en las marcas de	86400000 (24 * 60 * 60 * 1000 ms, es decir, 1 día)

Propiedad	Descripción	Valor predeterminado
	tiempo supera este umbral especificado. Esta configuración se ignora si. `log.message.timestamp.type=LogAppendTime`	

Propiedad	Descripción	Valor predeterminado
log.message.timestamp.before.max.ms	La diferencia de marca de tiempo permitida entre la marca de tiempo del corredor y la marca de tiempo del mensaje. La marca de tiempo del mensaje puede ser anterior o igual a la marca de tiempo del corredor, con la diferencia máxima permitida determinada por el valor establecido en esta configuración. Si `log.message.timestamp.type=CreateTime`, el mensaje se rechazará si la diferencia en las marcas de	86400000 (24 * 60 * 60 * 1000 ms, es decir, 1 día)

Propiedad	Descripción	Valor predeterminado
	tiempo supera este umbral especificado. Esta configuración se ignora si. `log.message.timestamp.type=LogAppendTime`	
log.message.timestamp.type	Especifica si la marca temporal del mensaje es la hora de creación del mensaje o la hora de adición del registro. Los valores permitidos son `CreateTime` y `LogAppendTime` .	Apache Kafka es el valor predeterminado
log.retention.bytes	Tamaño máximo del registro antes de eliminarlo.	Apache Kafka predeterminado

Propiedad	Descripción	Valor predeterminado
log.retention.ms	Número de milisegundos que se debe conservar un archivo de registro antes de eliminarlo.	Apache Kafka es el predeterminado
max.connection.creation.rate	La velocidad máxima de creación de conexiones permitida por el bróker en cualquier momento.	Apache Kafka es el predeterminado
número máximo de conexiones	El número máximo de conexiones permitidas en el bróker en cualquier momento. Este límite se aplica además de cualquier límite por IP configurado mediante`max.connections.per.ip` .	Apache Kafka es el predeterminado

Propiedad	Descripción	Valor predeterminado
max.connections.per.ip	El número máximo de conexiones permitidas desde cada dirección IP. Se puede configurar `0` si hay anulaciones configuradas mediante la propiedad max.connections.per.ip.overrides. Si se alcanza el límite, se eliminan las conexiones nuevas desde la dirección IP.	Apache Kafka es el valor predeterminado

Propiedad	Descripción	Valor predeterminado
max.connections.per.ip.overrides	Una lista separada por comas por IP o nombre de host reemplaza el número máximo de conexiones predeterminado. Un ejemplo de valor es `hostName:100,127.0.0.1:200`	El valor predeterminado de Apache Kafka

Propiedad	Descripción	Valor predeterminado
message.max.bytes	Tamaño de lote de registros más grande que admite Kafka. Si aumenta este valor y hay consumidores anteriores a 0.10.2, también debe aumentar el tamaño de recuperación de los consumidores para que se puedan recuperar lotes de registros de este tamaño. La versión de formato de mensaje más reciente siempre agrupa los mensajes en lotes para aumentar la eficacia. Las versiones de formato de mensaje anteriores no agrupan los	Apache Kafka es el predeterminado

Propiedad	Descripción	Valor predeterminado
	registros sin comprimir en lotes y, en este caso, este límite se aplica únicament e a un solo registro. Puede establecer este valor por tema con la `max.messa ge.bytes` configuración a nivel de tema.	

Propiedad	Descripción	Valor predeterminado
producer.id.expiration.ms	El tiempo en ms que esperará el líder de una partición temática antes de que el productor caduque. IDs El productor IDs no caducará mientras una transacción asociada a él esté en curso. Ten en cuenta que el productor IDs puede caducar antes si se elimina la última escritura del identificador de productor debido a la configuración de retención del tema. Establecer este valor igual o superior al anterior `delivery.`	Apache Kafka es el valor predeterminado

Propiedad	Descripción	Valor predeterminado
	`timeout.ms` puede ayudar a evitar la caducidad durante los reintentos y a evitar la duplicación de mensajes, pero el valor predeterminado debería ser razonable para la mayoría de los casos de uso.	

Configuraciones de nivel temático en Express Brokers

Puede utilizar los comandos de Apache Kafka para establecer o modificar propiedades de configuración de nivel de tema para temas nuevos y existentes. Si no puede proporcionar ninguna configuración a nivel de tema, Amazon MSK usa la configuración predeterminada del bróker. Al igual que con las configuraciones a nivel de intermediario, Amazon MSK protege algunas de las propiedades de configuración de nivel de tema contra cambios. Los ejemplos incluyen el factor de replicación y. `min.insync.replicas` `unclean.leader.election.enable` Si intenta crear un tema con un valor de factor de replicación distinto de`3`, Amazon MSK creará el tema con un factor de replicación `3` predeterminado. Para obtener más información acerca de las propiedades de configuración de temas y ejemplos de cómo establecerlas, consulte Topic-Level Configs en la documentación de Apache Kafka.

Propiedad	Descripción
cleanup.policy	Esta configuración designa la política de retención que se utilizará en los segmentos de registro. La política de «eliminar» (que es

Propiedad	Descripción
	la predeterminada) descartará los segmentos antiguos cuando se alcance su límite de tiempo de retención o tamaño. La política de «compactación» permitirá la compactación de registros, que conserva el valor más reciente de cada clave. También es posible especific ar ambas políticas en una lista separada por comas (por ejemplo, «eliminar, compactar »). En este caso, los segmentos antiguos se descartarán según la configuración de tiempo y tamaño de retención, mientras que los segmentos retenidos se compactarán. La compactación en los corredores Express se activa cuando los datos de una partición alcanzan los 256 MB.
compression.type	Especifique el tipo de compresión final para un tema determinado. Esta configuración acepta los códecs de compresión estándar (`gzip`,, `snappy` `lz4`,`zstd`). Además, acepta `uncompressed` lo que equivale a no comprimir, es decir`producer`, conservar el códec de compresión original establecido por el productor.

Propiedad	Descripción
eliminar.retention.ms	El tiempo que se debe conservar los marcadores de lápidas borrados para los temas compactados en registros. Esta configuración también establece un límite en el tiempo que un consumidor debe completar una lectura si comienza con un offset de 0 para garantiza r que obtiene una instantánea válida de la etapa final. De lo contrario, es posible que se recopilen lápidas eliminadas antes de que terminen de escanearlas. El valor predeterminado de esta configuración es 86400000 (24 * 60 * 60 * 1000 ms, es decir, 1 día), el valor predeterminado de Apache Kafka
max.message.bytes	El mayor tamaño de lote de registros permitido por Kafka (después de la compresión, si la compresión está habilitada). Si esta cifra aumenta y hay consumidores mayores de esa edad`0.10.`2, también se debe aumentar el tamaño de búsqueda de los consumido res para que puedan obtener lotes récord de este tamaño. En la versión de formato de mensaje más reciente, los registros siempre se agrupan en lotes para aumentar la eficacia. En versiones de formato de mensaje anteriore s, los registros sin comprimir no se agrupan en lotes y este límite se aplica únicamente a un solo registro en este caso. Esto se puede configurar por tema con el nivel de tema. `max.message.bytes config`

Propiedad	Descripción
message.timestamp.after.max.ms	Esta configuración establece la diferencia de marca de tiempo permitida entre la marca de tiempo del mensaje y la marca de tiempo del corredor. La marca de tiempo del mensaje puede ser posterior o igual a la marca de tiempo del corredor, con la diferencia máxima permitida determinada por el valor estableci do en esta configuración. Si`message.t imestamp.type=CreateTime` , el mensaje se rechazará si la diferencia en las marcas de tiempo supera este umbral especificado. Esta configuración se ignora si. `message.timestamp.type=LogA ppendTime`
message.timestamp.before.max.ms	Esta configuración establece la diferencia de marca de tiempo permitida entre la marca de tiempo del corredor y la marca de tiempo del mensaje. La marca de tiempo del mensaje puede ser anterior o igual a la marca de tiempo del corredor, con la diferencia máxima permitida determinada por el valor estableci do en esta configuración. Si`message.t imestamp.type=CreateTime` , el mensaje se rechazará si la diferencia en las marcas de tiempo supera este umbral especificado. Esta configuración se ignora si. `message.timestamp.type=LogA ppendTime`
message.timestamp.type	Defina si la marca de tiempo del mensaje es la hora de creación del mensaje o la hora de adición del registro. El valor debe ser uno o `CreateTime` `LogAppendTime`

Propiedad	Descripción
min.compaction.lag.ms	El tiempo mínimo que un mensaje permanecerá sin compactar en el registro. Esta configuración solo se aplica a los registros que se están compactando. El valor predeterminado de esta configuración es 0, es el valor predeterminado de Apache Kafka
max.compaction.lag.ms	El tiempo máximo que un mensaje permanecerá inapto para compactarse en el registro. Esta configuración solo se aplica a los registros que se están compactando. Esta configuración estaría limitada en el rango de [7 días, Long.Max]. El valor predeterminado de esta configuración es 9223372036854775807, el valor predeterminado de Apache Kafka.

Propiedad	Descripción
retention.bytes	Esta configuración controla el tamaño máximo que puede alcanzar una partición (que consta de segmentos de registro) antes de descartar los segmentos de registro antiguos para liberar espacio si utilizamos la política de retención de «eliminar». De forma predeterminada, no hay límite de tamaño, solo un límite de tiempo. Como este límite se aplica a nivel de partición, multiplíquelo por el número de particiones para calcular la retención del tema en bytes. Además, `retention.bytes configuration` funciona independi entemente de `segment.ms` las `segment.bytes` configuraciones. Además, activa la rotación de un nuevo segmento si `retention.bytes` está configurado a cero.
retention.ms	Esta configuración controla el tiempo máximo que conservaremos un registro antes de descartar los segmentos de registro antiguos para liberar espacio si utilizamos la política de retención de «eliminar». Esto represent a un acuerdo de nivel de servicio sobre la rapidez con la que los consumidores deben leer sus datos. Si se establece en `-1`, no se aplica ningún límite de tiempo. Además, `retention.ms` la configuración funciona de forma independiente de `segment.ms` las `segment.bytes` configuraciones. Además, activa la rotación de un nuevo segmento si se cumple la `retention.ms` condición.

Configuraciones de solo lectura de Express Brokers

Amazon MSK establece los valores de estas configuraciones y las protege de los cambios que puedan afectar a la disponibilidad del clúster. Estos valores pueden cambiar en función de la versión de Apache Kafka que se ejecute en el clúster, así que recuerde comprobar los valores de su clúster específico. Estos son algunos ejemplos.

Configuraciones de solo lectura de Express Brokers

Propiedad	Descripción	Valor de Express Broker
broker.id	El identificador del corredor de este servidor.	1,2,3...
broker.rack	Rack of the broker. Se utilizará en la asignación de replicaciones con reconocimiento de racks para garantizar la tolerancia a errores. Ejemplos: `RACK1`, `us-east-1d`	ID de AZ o ID de subred
default.replication.factor	Factores de replicación predeterminados para todos los temas.	3
fetch.max.bytes	El número máximo de bytes que devolveremos para una solicitud de recuperación.	Apache Kafka es el predeterminado
tamaño máximo del grupo	El número máximo de consumidores que puede alojar un solo grupo de consumidores.	Apache Kafka (predeterminado)
inter.broker.listener.name	Nombre del oyente utilizado para la comunicación entre los corredores.	REPLICATION_SECURE o REPLICATION

Propiedad	Descripción	Valor de Express Broker
inter.broker.protocol.version	Especifica qué versión del protocolo entre corredores se utiliza.	Apache Kafka es el valor predeterminado
oyentes	Lista de oyentes: lista separada por comas de URIs lo que escucharemos y los nombres de los oyentes. Puede establecer la propiedad `advertised.listeners property` , pero no la propiedad. `listeners`	Generado por MSK
log.message.format.versión	Especifique la versión del formato de mensaje que el agente utilizará para anexar los mensajes a los registros.	Apache Kafka es el valor predeterminado

Propiedad	Descripción	Valor de Express Broker
min.insync.replicas	Cuando un productor establece acks en `all` (o-1), el valor en `min.insyn c.replicas` especifica el número mínimo de réplicas que deben reconocer una escritura para que la escritura se considere correcta. Si no se puede alcanzar este mínimo, el productor hace una excepción (una de las dos opciones `NotEnough Replicas NotEnough ReplicasAfterAppen d`). Puede utilizar el valor de los paquetes de su fabricante para reforzar las garantías de durabilidad. Estableciendo los paquetes en «Todos». Esto garantiza que el productor emita una excepción si la mayoría de las réplicas no reciben una escritura.	2
num.io.threads	Número de subprocesos que el servidor utiliza para generar solicitudes, que pueden incluir E/S de disco. (m7g.large, 8), (m7g.xlarge, 8), (m7g.2xla rge, 16), (m7g.4xlarge, 32), (m7g.8xlarge, 64), (m7g.12xl arge, 96), (m7g.16xlarge, 128)	Según el tipo de instancia. =Math.max (8, 2 * v) CPUs

Propiedad	Descripción	Valor de Express Broker
num.network.threads	Número de subprocesos que el servidor utiliza para recibir solicitudes de la red y enviar respuestas a la red. (m7g.large, 8), (m7g.xlarge, 8), (m7g.2xlarge, 8), (m7g.4xlarge, 16), (m7g.8xlarge, 32), (m7g.12xlarge, 48), (m7g.16xlarge, 64)	Basado en el tipo de instancia. =Math.max (8, v) CPUs
replica.fetch.response.max.bytes	El número máximo de bytes previsto para la respuesta de recuperación completa. Los registros se recuperan en lotes y, si el primer lote de registro de la primera partición que no está vacía de la recuperación es superior a este valor, el lote de registro se devolverá para asegurar que se lleva a cabo el progreso. Esto no es un valor máximo absoluto. Las propiedades `message.max.bytes` (broker config) o `max.message.bytes` (topic config) especifican el tamaño máximo de lote de registros que acepta el bróker.	Apache Kafka es el valor predeterminado

Propiedad	Descripción	Valor de Express Broker
request.timeout.ms	La configuración controla la cantidad máxima de tiempo que el cliente esperará la respuesta de una solicitud. Si la respuesta no se recibe antes de que se agote el tiempo de espera, el cliente volverá a enviar la solicitud si es necesario o rechazará la solicitud si se agotan los reintentos.	Apache Kafka es el predeterminado
transaction.state.log.min.isr	`min.insync.replicas` Configuración anulada para el tema de la transacción.	2
transaction.state.log.replication.factor	El factor de replicación del tema de transacción.	Apache Kafka (predeterminado)
unclean.leader.election.enable	Permite que las réplicas que no estén incluidas en el ISR actúen como líderes como último recurso, aunque esto pueda provocar la pérdida de datos.	FALSO

Operaciones de configuración del bróker

Las configuraciones de los brókers de Apache Kafka son estáticas o dinámicas. Las configuraciones estáticas requieren el reinicio del broker para poder aplicar la configuración. Las configuraciones dinámicas no necesitan reiniciar el broker para que la configuración se actualice. Para obtener más información sobre las propiedades de configuración y los modos de actualización, consulte Configuración de Apache Kafka.

En este tema se describe cómo crear configuraciones de MSK personalizadas y cómo realizar operaciones en ellas. Para obtener información acerca de cómo utilizar las configuraciones de MSK para crear o actualizar clústeres, consulte the section called “Características y conceptos clave”.

Temas

- Crear una configuración
- Actualización de la configuración
- Elimine la configuración
- Obtenga los metadatos de configuración
- Obtenga detalles sobre la revisión de la configuración
- Enumere las configuraciones de su cuenta para la región actual
- Estados de configuración de Amazon MSK

Crear una configuración

En este proceso, se describe cómo crear una configuración personalizada de Amazon MSK y cómo utilizarlo para realizar operaciones.

1. Cree un archivo donde especifique las propiedades de configuración que desea establecer y los valores que desea asignarles. A continuación se muestra el contenido de un archivo de configuración de ejemplo.

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

log.roll.ms = 604800000
```

2. Ejecute el siguiente AWS CLI comando y *config-file-path* sustitúyalo por la ruta del archivo en el que guardó la configuración en el paso anterior.

> **Note**
>
> El nombre que elija para la configuración debe coincidir con la siguiente expresión regular: «^[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
```

El siguiente es un ejemplo de una respuesta correcta después de ejecutar este comando.

```
{
    "Arn": "arn:aws:kafka:us-east-1:123456789012:configuration/SomeTest/
abcdabcd-1234-abcd-1234-abcd123e8e8e-1",
    "CreationTime": "2019-05-21T19:37:40.626Z",
    "LatestRevision": {
        "CreationTime": "2019-05-21T19:37:40.626Z",
        "Description": "Example configuration description.",
        "Revision": 1
    },
    "Name": "ExampleConfigurationName"
}
```

3. El comando anterior devuelve un nombre de recurso de Amazon (ARN) para la nueva configuración. Guarde este ARN, porque lo necesita para hacer referencia a esta configuración en otros comandos. Si pierde el ARN de la configuración, puede enumerar todas las configuraciones de su cuenta para volver a encontrarlo.

Actualización de la configuración

En este proceso, se describe cómo actualizar una configuración de Amazon MSK personalizada.

1. Cree un archivo donde especifique las propiedades de configuración que desea actualizar y los valores que desea asignarles. A continuación se muestra el contenido de un archivo de configuración de ejemplo.

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

min.insync.replicas = 2
```

2. Ejecute el siguiente AWS CLI comando y *config-file-path* sustitúyalo por la ruta del archivo en el que guardó la configuración en el paso anterior.

*configuration-arn*Sustitúyalo por el ARN que obtuvo al crear la configuración. Si no guardó el ARN cuando creó la configuración, puede usar el comando `list-configurations` para enumerar todas las configuraciones de su cuenta. La configuración que desea incluir en la lista aparece en la respuesta. El ARN de la configuración también aparece en dicha lista.

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

3. El siguiente es un ejemplo de una respuesta correcta después de ejecutar este comando.

```
{
    "Arn": "arn:aws:kafka:us-east-1:123456789012:configuration/SomeTest/
abcdabcd-1234-abcd-1234-abcd123e8e8e-1",
    "LatestRevision": {
        "CreationTime": "2020-08-27T19:37:40.626Z",
        "Description": "Example configuration revision description.",
        "Revision": 2
    }
}
```

Elimine la configuración

En el siguiente procedimiento se muestra cómo eliminar una configuración que no está asociada a un clúster. No puede eliminar una configuración asociada a un clúster.

1. Para ejecutar este ejemplo, *configuration-arn* sustitúyalo por el ARN que obtuvo al crear la configuración. Si no guardó el ARN cuando creó la configuración, puede usar el comando `list-configurations` para enumerar todas las configuraciones de su cuenta. La configuración que desea incluir en la lista aparece en la respuesta. El ARN de la configuración también aparece en dicha lista.

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

2. El siguiente es un ejemplo de una respuesta correcta después de ejecutar este comando.

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

```
}
```

Obtenga los metadatos de configuración

En el siguiente procedimiento, se muestra cómo describir una configuración de Amazon MSK para obtener metadatos sobre la configuración.

1. Este comando devuelve metadatos acerca de la configuración. Para obtener una descripción detallada de la configuración, ejecute `describe-configuration-revision`.

   Para ejecutar este ejemplo, *`configuration-arn`* sustitúyalo por el ARN que obtuvo al crear la configuración. Si no guardó el ARN cuando creó la configuración, puede usar el comando `list-configurations` para enumerar todas las configuraciones de su cuenta. La configuración que desea incluir en la lista aparece en la respuesta. El ARN de la configuración también aparece en dicha lista.

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

2. El siguiente es un ejemplo de una respuesta correcta después de ejecutar este comando.

   ```
   {
       "Arn": "arn:aws:kafka:us-east-1:123456789012:configuration/SomeTest/abcdabcd-
   abcd-1234-abcd-abcd123e8e8e-1",
       "CreationTime": "2019-05-21T00:54:23.591Z",
       "Description": "Example configuration description.",
       "KafkaVersions": [
           "1.1.1"
       ],
       "LatestRevision": {
           "CreationTime": "2019-05-21T00:54:23.591Z",
           "Description": "Example configuration description.",
           "Revision": 1
       },
       "Name": "SomeTest"
   }
   ```

Obtenga detalles sobre la revisión de la configuración

En este proceso, se ofrece una descripción detallada de la revisión de la configuración de Amazon MSK.

Si usa el comando `describe-configuration` para describir una configuración de MSK, verá los metadatos de la configuración. Para obtener una descripción de la configuración, use el comando `describe-configuration-revision`.

- Ejecute el siguiente comando y *`configuration-arn`* sustitúyalo por el ARN que obtuvo al crear la configuración. Si no guardó el ARN cuando creó la configuración, puede usar el comando `list-configurations` para enumerar todas las configuraciones de su cuenta. La configuración que desea incluir en la lista que aparece en la respuesta. El ARN de la configuración también aparece en dicha lista.

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

  El siguiente es un ejemplo de una respuesta correcta después de ejecutar este comando.

  ```
  {
      "Arn": "arn:aws:kafka:us-east-1:123456789012:configuration/SomeTest/abcdabcd-
  abcd-1234-abcd-abcd123e8e8e-1",
      "CreationTime": "2019-05-21T00:54:23.591Z",
      "Description": "Example configuration description.",
      "Revision": 1,
      "ServerProperties":
   "YXV0by5jcmVhdGUudG9waWNzLmVuYWJsZSA9IHRydWUKCgp6b29rZWVwZXIuY29ubmVjdGlvbi50aW1lb3V0Lm1zI
  }
  ```

  El valor de `ServerProperties` se codifica en base64. Si utiliza un decodificador base64 (por ejemplo, https://www.base64decode.org/) para decodificarlo manualmente, obtendrá el contenido del archivo de configuración original que utilizó para crear la configuración personalizada. En este caso, obtiene lo siguiente:

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

  log.roll.ms = 604800000
  ```

Enumere las configuraciones de su cuenta para la región actual

Este proceso describe cómo enumerar todas las configuraciones de Amazon MSK de su cuenta para la AWS región actual.

- Ejecute el siguiente comando.

```
aws kafka list-configurations
```

El siguiente es un ejemplo de una respuesta correcta después de ejecutar este comando.

```
{
    "Configurations": [
        {
            "Arn": "arn:aws:kafka:us-east-1:123456789012:configuration/SomeTest/
abcdabcd-abcd-1234-abcd-abcd123e8e8e-1",
            "CreationTime": "2019-05-21T00:54:23.591Z",
            "Description": "Example configuration description.",
            "KafkaVersions": [
                "1.1.1"
            ],
            "LatestRevision": {
                "CreationTime": "2019-05-21T00:54:23.591Z",
                "Description": "Example configuration description.",
                "Revision": 1
            },
            "Name": "SomeTest"
        },
        {
            "Arn": "arn:aws:kafka:us-east-1:123456789012:configuration/SomeTest/
abcdabcd-1234-abcd-1234-abcd123e8e8e-1",
            "CreationTime": "2019-05-03T23:08:29.446Z",
            "Description": "Example configuration description.",
            "KafkaVersions": [
                "1.1.1"
            ],
            "LatestRevision": {
                "CreationTime": "2019-05-03T23:08:29.446Z",
                "Description": "Example configuration description.",
                "Revision": 1
            },
            "Name": "ExampleConfigurationName"
        }
    ]
}
```

Estados de configuración de Amazon MSK

Una configuración de Amazon MSK puede tener uno de los siguientes estados: Para realizar una operación en una configuración, la configuración debe estar en el estado `ACTIVE` o `DELETE_FAILED`:

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

Aplicación de parches

Aplicación de parches en clústeres aprovisionados por MSK

Amazon MSK actualiza periódicamente el software de los agentes de su aprovisionado. El mantenimiento incluye las actualizaciones planificadas o las reparaciones no planificadas. El mantenimiento planificado incluye las actualizaciones del sistema operativo, las actualizaciones de seguridad y otras actualizaciones de software necesarias para mantener el estado, la seguridad y el rendimiento del clúster. Realizamos un mantenimiento no planificado para solucionar la degradación repentina de la infraestructura. Realizamos el mantenimiento en los corredores Standard y Express, pero las experiencias son diferentes.

Aplicación de parches para corredores estándar

Las actualizaciones de sus agentes aprovisionado no tienen ningún impacto en la escritura y lectura de sus aplicaciones.

Amazon MSK utiliza actualizaciones de software continuas para mantener una alta disponibilidad de sus clústeres. Durante este proceso, los agentes se reinician de uno en uno y Kafka traslada automáticamente el liderazgo a otro agente en línea. Los clientes de Kafka disponen de mecanismos integrados para detectar automáticamente el cambio de dirección de las particiones y seguir escribiendo y leyendo los datos en un clúster de MSK. Siga las instrucciones Prácticas recomendadas para los clientes de Apache Kafka para que su clúster funcione sin problemas en todo momento, incluso durante la aplicación de parches.

Tras la desconexión de un agente, es normal que sus clientes cometan errores transitorios de desconexión. También observará durante un breve periodo (hasta 2 minutos, normalmente menos)

algunos picos en la latencia de lectura y escritura del p99 (normalmente altos milisegundos, hasta aproximadamente 2 segundos). Estos picos son esperados y se deben a que el cliente vuelve a conectarse a un nuevo agente líder; no afectan en sus productos ni en su consumo y se resolverán al volver a conectarse. Para obtener más información, consulte El agente no está en línea y el cliente realiza una conmutación por error.

También observará un aumento en la métrica`UnderReplicatedPartitions`, lo cual es de esperar, ya que las particiones del agente que se cerró ya no replican datos. Esto no afecta a las escrituras y lecturas de las aplicaciones, ya que las réplicas de estas particiones alojadas en otros agentes ahora atienden las solicitudes.

Tras la actualización del software, cuando el agente vuelva a estar en línea, tendrá que “ponerse al día” con los mensajes producidos mientras estaba fuera de línea. Durante la recuperación, también puede observar un aumento en el uso del rendimiento del volumen y de la CPU. Esto no debería afectar a las escrituras y lecturas del clúster si sus agentes disponen de suficientes recursos de CPU, memoria, red y volumen.

Parches para corredores de Express

No hay períodos de mantenimiento para los corredores de Express. Amazon MSK actualiza automáticamente su clúster de forma continua y distribuida en el tiempo, lo que significa que puede esperar reinicios ocasionales y singulares de los corredores a lo largo del mes. Esto garantiza que no tendrá que hacer planes ni ajustes en torno a períodos de mantenimiento únicos para todo el clúster. Como siempre, el tráfico se mantendrá ininterrumpido durante el reinicio de los corredores, ya que el liderazgo pasará a manos de otros corredores que seguirán atendiendo las solicitudes.

Los corredores Express vienen configurados con las mejores prácticas y barreras que hacen que su clúster sea resistente a los cambios de carga que puedan producirse durante el mantenimiento. Amazon MSK establece cuotas de rendimiento para sus corredores de Express para mitigar el impacto de la sobrecarga de su clúster, lo que puede provocar problemas durante el reinicio del corredor. Estas mejoras eliminan la necesidad de notificaciones anticipadas y de plazos de planificación y mantenimiento cuando se utiliza Express Brokers.

Los corredores de Express siempre replican sus datos de tres maneras para que sus clientes se conmuten automáticamente por error durante los reinicios. No tiene que preocuparse de que los temas no estén disponibles debido a que el factor de replicación está establecido en 1 o 2. Además, ponerse al día con un corredor Express que se reinicia es más rápido que con los corredores estándar. La mayor velocidad de aplicación de parches en los corredores de Express significa que

cualquier actividad del plano de control que haya programado para su clúster tendrá una interrupción mínima de la planificación.

Como ocurre con todas las aplicaciones de Apache Kafka, los clientes que se conectan a los corredores de Express mantienen un contrato cliente-servidor compartido. Sigue siendo fundamental configurar a sus clientes para que gestionen el cambio de liderazgo entre los corredores. Siga las instrucciones Prácticas recomendadas para los clientes de Apache Kafka para que su clúster funcione sin problemas en todo momento, incluso durante la aplicación de parches. Tras el reinicio de un agente, es normal que sus clientes cometan errores transitorios de desconexión. Esto no afectará a su producción y consumo, ya que los corredores seguidores asumirán el liderazgo de la partición. Sus clientes de Apache Kafka cambiarán automáticamente por error y empezarán a enviar solicitudes a los nuevos agentes líderes.

El agente no está en línea y el cliente realiza una conmutación por error

Kafka permite utilizar un agente que no está en línea; un único agente que no está en línea en un clúster sano y equilibrado que siga las prácticas recomendadas no se verá afectado ni provocará fallos en la producción o el consumo. Esto se debe a que otro agente asumirá el liderazgo de la partición y a que la cartera de clientes de Kafka cambiará automáticamente por error y empezará a enviar solicitudes a los nuevos agentes líderes.

Contrato cliente-servidor

Esto se traduce en un contrato compartido entre la cartera de clientes y el comportamiento del servidor; el servidor debe asignar correctamente uno o más nuevos líderes y el cliente debe cambiar de agente para enviar las solicitudes a los nuevos líderes en el momento oportuno.

Kafka utiliza excepciones para controlar este flujo:

Un procedimiento de ejemplo

1. El agente A entra en estado de desconexión.
2. El cliente de Kafka recibe una excepción (normalmente desconectado de la red o not_leader_for_partition).
3. Estas excepciones hacen que el cliente de Kafka actualice sus metadatos para conocer a los líderes más recientes.
4. El cliente de Kafka vuelve a enviar solicitudes a los nuevos líderes de partición a través de otros agentes.

Este proceso suele tardar menos de 2 segundos con el cliente Java vendido y las configuraciones predeterminadas. Los errores del cliente son detallados y repetitivos, pero no son motivo de preocupación, como se indica en el nivel “WARN”.

Ejemplo: excepción 1

```
10:05:25.306 [kafka-producer-network-thread | producer-1] WARN
o.a.k.c.producer.internals.Sender - [Producer clientId=producer-1] Got
error produce response with correlation id 864845 on topic-partition
msk-test-topic-1-0, retrying (2147483646 attempts left). Error:
NETWORK_EXCEPTION. Error Message: Disconnected from node 2
```

Ejemplo: excepción 2

```
10:05:25.306 [kafka-producer-network-thread | producer-1] WARN
o.a.k.c.producer.internals.Sender - [Producer clientId=producer-1] Received
invalid metadata error in produce request on partition msk-test-topic-1-41
due to org.apache.kafka.common.errors.NotLeaderOrFollowerException: For
requests intended only for the leader, this error indicates that the broker
is not the current leader. For requests intended for any replica, this
error indicates that the broker is not a replica of the topic partition..
Going to request metadata update now"
```

Los clientes de Kafka resolverán automáticamente estos errores, normalmente en 1 segundo y como máximo en 3 segundos. Esto se presenta como produce/consume una latencia de p99 en las métricas del lado del cliente (por lo general, una latencia alta en milisegundos a los 100). Un valor superior a este valor suele indicar un problema con la configuración del cliente o con la carga del controlador del servidor. Consulte la sección de solución de problemas.

Una conmutación por error exitosa puede verificarse comprobando el aumento de las métricas de `BytesInPerSec` y `LeaderCount` de otros agentes, lo que demuestra que el tráfico y el liderazgo se han movido como se esperaba. También observará un aumento en la métrica `UnderReplicatedPartitions`, lo que se espera cuando las réplicas no están en línea con el agente de cierre.

Solución de problemas

El flujo anterior puede interrumpirse si se rompe el contrato cliente-servidor. Los motivos más comunes del problema son:

- Configuración incorrecta o uso incorrecto de las carteras de clientes de Kafka.
- Comportamientos predeterminados inesperados y errores en las carteras de clientes de terceros.
- Sobrecarga del controlador que hace que la asignación del líder de partición sea más lenta.
- Elección de un nuevo controlador que hace que la asignación del líder de partición sea más lenta.

Con el fin de garantizar un comportamiento correcto a la hora de gestionar los fallos de liderazgo, recomendamos:

- Se deben seguir las prácticas recomendadas del servidor para garantizar que el agente de controladores tenga la escala adecuada para evitar una lenta asignación de liderazgo.
- Las carteras de clientes deben tener habilitados los reintentos para garantizar que el cliente gestione la conmutación por error.
- Las bibliotecas cliente deben tener configurado retry.backoff.ms (100 por defecto) para evitar tormentas. connection/request
- Las carteras de clientes deben configurar request.timeout.ms y delivery.timeout.ms en valores acordes con el SLA de las aplicaciones. Los valores más altos harán que la conmutación por error sea más lenta para algunos tipos de errores.
- Las carteras de clientes deben asegurarse de que bootstrap.servers contenga al menos 3 agentes aleatorios para evitar que la disponibilidad se vea afectada en el momento de la detección inicial.
- Algunas carteras de clientes son de nivel inferior a otras y esperan que el desarrollador de la aplicación implemente por sí mismo la lógica de reintentos y la gestión de excepciones. Consulte la documentación específica de la biblioteca de clientes para ver, por ejemplo, el uso, y asegúrese de seguir la lógica correcta. reconnect/retry
- Recomendamos supervisar la latencia del cliente para comprobar la producción, el recuento de solicitudes satisfactorias y el recuento de errores en caso de errores que no se puedan volver a intentar.
- Hemos observado que las antiguas carteras de golang y ruby de terceros permanecen detalladas durante todo el periodo de inactividad del agente, a pesar de que las solicitudes de producción y consumo no se ven afectadas. Le recomendamos que siempre supervise las métricas a nivel empresarial, además de solicitar métricas de éxito y errores, para determinar si hay un impacto real o un ruido en sus registros.
- Los clientes no deberían alarmarse ante las excepciones transitorias en el caso de network/not_leader, ya que son normales, no impactan y se espera que formen parte del protocolo Kafka.

- Los clientes no deberían activar las alarmas, UnderReplicatedPartitions ya que son normales, no impactan y se espera que se produzcan cuando se trata de un intermediario fuera de línea.

Registro de Amazon MSK

Puede entregar los registros del broker de Apache Kafka a uno o más de los siguientes tipos de destinos: Amazon CloudWatch Logs, Amazon S3, Amazon Data Firehose. También puede registrar las llamadas a la API de Amazon MSK con AWS CloudTrail.

> **Note**
>
> Los registros de los corredores no están disponibles en los corredores de Express.

Registros de agente

Los registros de agente le permiten solucionar problemas de las aplicaciones de Apache Kafka y analizar las comunicaciones con su clúster de MSK. Puede configurar su clúster de MSK nuevo o existente para entregar registros de corredores de nivel de información a uno o más de los siguientes tipos de recursos de destino: un grupo de CloudWatch registros, un depósito de S3 o un flujo de entrega de Firehose. A través de Firehose, puede enviar los datos de registro de su flujo de entrega a OpenSearch Service. Debe crear un recurso de destino antes de configurar el clúster para enviarle los registros de los agentes. Amazon MSK no crea estos recursos de destino si aún no existen. Para obtener información acerca de estos tres tipos de recursos de destino y cómo crearlos, consulte la siguiente documentación:

- Amazon CloudWatch Logs
- Amazon S3
- Amazon Data Firehose

Permisos necesarios

Para configurar un destino para los registros de los agentes de Amazon MSK, la identidad de IAM que utilice para las acciones de Amazon MSK debe tener los permisos descritos en la política AWS política gestionada: Amazon MSKFull Access.

Para transmitir registros de agente a un bucket de S3, también necesita el permiso `s3:PutBucketPolicy`. Para obtener información acerca de las políticas de bucket de S3, consulte

¿Cómo agrego una política de bucket de S3? en la Guía del usuario de Amazon S3. Para obtener información acerca de las políticas de IAM en general, consulte Administración de accesos en la Guía del usuario de IAM.

Política de claves de KMS necesarias para usar con buckets de SSE-KMS

Si habilitó el cifrado del lado del servidor para su bucket de S3 mediante claves AWS KMS administradas (SSE-KMS) con una clave administrada por el cliente, añada lo siguiente a la política de claves de su clave de KMS para que Amazon MSK pueda escribir archivos de broker en el bucket.

```
{
  "Sid": "Allow Amazon MSK to use the key.",
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "delivery.logs.amazonaws.com"
    ]
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

Configure los registros de los corredores mediante el AWS Management Console

Si va a crear un nuevo clúster, busque el encabezado Broker log delivery (Entrega de registros de agente) en la sección Monitoring (Monitoreo). Puede especificar los destinos donde desea que Amazon MSK entregue los registros de agente.

Para un clúster existente, elija el clúster de la lista de clústeres y, a continuación, elija la pestaña Propiedades. Desplácese hacia abajo hasta la sección Envío de registros y, a continuación, elija el botón Editar. Puede especificar los destinos donde desea que Amazon MSK entregue los registros de agente.

Configure los registros de los corredores mediante el AWS CLI

Si utiliza los comandos `update-monitoring` o `create-cluster`, tiene la opción de especificar el parámetro `logging-info` y pasarlo a una estructura JSON como en el siguiente ejemplo. En este JSON, los tres tipos de destino son opcionales.

```
{
  "BrokerLogs": {
    "S3": {
      "Bucket": "amzn-s3-demo-bucket",
      "Prefix": "ExamplePrefix",
      "Enabled": true
    },
    "Firehose": {
      "DeliveryStream": "ExampleDeliveryStreamName",
      "Enabled": true
    },
    "CloudWatchLogs": {
      "Enabled": true,
      "LogGroup": "ExampleLogGroupName"
    }
  }
}
```

Configuración de los registros de agente con la API

Puede especificar la `loggingInfo` estructura opcional en el JSON que va a pasar a las UpdateMonitoringoperaciones CreateClustero.

> **Note**
>
> De forma predeterminada, cuando el registro de agentes está habilitado, Amazon MSK almacena los registros de `INFO` en los destinos especificados. Sin embargo, los usuarios de las versiones 2.4.X y posteriores de Apache Kafka pueden establecer dinámicamente el nivel de registros de agente en cualquiera de los niveles de registro log4j. Para obtener información sobre cómo configurar dinámicamente el nivel de registros de agente, consulte KIP-412: Extend Admin API to support dynamic application log levels. Si establece el nivel de registro de forma dinámica en `DEBUG` o `TRACE`, recomendamos que utilice Amazon S3 o Firehose como destino de los registros. Si utiliza CloudWatch Logs como destino de registros y habilita `DEBUG` o `TRACE` nivela el registro de forma dinámica, Amazon MSK puede entregar continuamente una muestra de registros. Esto puede afectar considerablemente

al rendimiento del agente y solo debe utilizarse cuando el nivel de registro `INFO` no sea lo suficientemente detallado como para determinar la causa raíz del problema.

Registre las llamadas a la API con AWS CloudTrail

> **Note**
>
> AWS CloudTrail los registros están disponibles para Amazon MSK solo cuando se utilizaControl de acceso de IAM.

Amazon MSK está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en Amazon MSK. CloudTrail captura las llamadas a la API como eventos. Las llamadas capturadas incluyen las llamadas desde la consola de Amazon MSK y las llamadas desde el código a las operaciones de la API de Amazon MSK. También captura las acciones de Apache Kafka, como la creación y modificación de temas y grupos.

Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos de Amazon MSK. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a Amazon MSK o la acción de Apache Kafka, la dirección IP desde la que se realizó la solicitud, quién la realizó, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, incluido cómo configurarlo y habilitarlo, consulte la Guía del AWS CloudTrail usuario.

Información de Amazon MSK en CloudTrail

CloudTrail está activado en su cuenta de Amazon Web Services al crear la cuenta. Cuando una actividad de eventos admitida se produce en un clúster de MSK, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puede ver, buscar y descargar los últimos eventos de la cuenta de Amazon Web Services. Para obtener más información, consulte Visualización de eventos con el historial de eventos de CloudTrail.

Para mantener un registro continuo de eventos en la cuenta de Amazon Web Services, incluidos los eventos de Amazon MSK, cree un registro de seguimiento. Un rastro permite CloudTrail entregar

archivos de registro a un bucket de Amazon S3. De manera predeterminada, cuando crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registro al bucket de Amazon S3 especificado. Además, puedes configurar otros servicios de Amazon para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- CloudTrail Servicios e integraciones compatibles
- Configuración de las notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de CloudTrail registro de varias regiones y recibir archivos de CloudTrail registro de varias cuentas

Amazon MSK registra todas las operaciones de Amazon MSK como eventos en archivos de CloudTrail registro. Además, registra las siguientes acciones de Apache Kafka.

- clúster kafka: DescribeClusterDynamicConfiguration
- cúmulo kafka: AlterClusterDynamicConfiguration
- cúmulo kafka: CreateTopic
- cúmulo kafka: DescribeTopicDynamicConfiguration
- cúmulo kafka: AlterTopic
- cúmulo kafka: AlterTopicDynamicConfiguration
- cúmulo kafka: DeleteTopic

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con credenciales de usuario root o de usuario AWS Identity and Access Management (IAM).
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro AWS servicio.

Para obtener más información, consulte el Elemento userIdentity de CloudTrail .

Ejemplo: entradas del archivo de registros de Amazon MSK

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que usted especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una solicitud única de cualquier fuente e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las llamadas a las API públicas y las acciones de Apache Kafka, por lo que no aparecen en ningún orden específico.

El siguiente ejemplo muestra las entradas de CloudTrail registro que muestran las acciones `DescribeCluster` y las de `DeleteCluster` Amazon MSK.


```
{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "ABCDEF0123456789ABCDE",
        "arn": "arn:aws:iam::012345678901:user/Joe",
        "accountId": "012345678901",
        "accessKeyId": "AIDACKCEVSQ6C2EXAMPLE",
        "userName": "Joe"
      },
      "eventTime": "2018-12-12T02:29:24Z",
      "eventSource": "kafka.amazonaws.com",
      "eventName": "DescribeCluster",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "aws-cli/1.14.67 Python/3.6.0 Windows/10 botocore/1.9.20",
      "requestParameters": {
        "clusterArn": "arn%3Aaws%3Akafka%3Aus-east-1%3A012345678901%3Acluster
%2Fexamplecluster%2F01234567-abcd-0123-abcd-abcd0123efa-2"
      },
      "responseElements": null,
      "requestID": "bd83f636-fdb5-abcd-0123-157e2fbf2bde",
      "eventID": "60052aba-0123-4511-bcde-3e18dbd42aa4",
      "readOnly": true,
      "eventType": "AwsApiCall",
      "recipientAccountId": "012345678901"
    },
    {
      "eventVersion": "1.05",
```

```
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "ABCDEF0123456789ABCDE",
        "arn": "arn:aws:iam::012345678901:user/Joe",
        "accountId": "012345678901",
        "accessKeyId": "AIDACKCEVSQ6C2EXAMPLE",
        "userName": "Joe"
      },
      "eventTime": "2018-12-12T02:29:40Z",
      "eventSource": "kafka.amazonaws.com",
      "eventName": "DeleteCluster",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "aws-cli/1.14.67 Python/3.6.0 Windows/10 botocore/1.9.20",
      "requestParameters": {
        "clusterArn": "arn%3Aaws%3Akafka%3Aus-east-1%3A012345678901%3Acluster
%2Fexamplecluster%2F01234567-abcd-0123-abcd-abcd0123efa-2"
      },
      "responseElements": {
        "clusterArn": "arn:aws:kafka:us-east-1:012345678901:cluster/
examplecluster/01234567-abcd-0123-abcd-abcd0123efa-2",
        "state": "DELETING"
      },
      "requestID": "c6bfb3f7-abcd-0123-afa5-293519897703",
      "eventID": "8a7f1fcf-0123-abcd-9bdb-1ebf0663a75c",
      "readOnly": false,
      "eventType": "AwsApiCall",
      "recipientAccountId": "012345678901"
    }
  ]
}
```


El siguiente ejemplo muestra una entrada de CloudTrail registro que demuestra la `kafka-cluster:CreateTopic` acción.


```
{
  "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"
}
```


Administración de metadatos

Amazon MSK admite los modos de administración de KRaft metadatos ZooKeeper o Apache.

A partir de la versión 3.7.x de Apache Kafka en Amazon MSK, puede crear clústeres que usen el modo en lugar del KRaft modo. ZooKeeper KRaftlos clústeres basados en sistemas basados en Kafka se basan en los controladores de Kafka para administrar los metadatos.

Temas

- ZooKeeper modo
- KRaft modo

ZooKeeper modo

Apache ZooKeeper es «un servicio centralizado para mantener la información de configuración, asignar nombres, proporcionar sincronización distribuida y proporcionar servicios grupales. Las

aplicaciones distribuidas utilizan todos estos tipos de servicios de una forma u otra", como Apache Kafka.

Si su clúster utiliza el ZooKeeper modo, puede seguir los pasos que se indican a continuación para obtener la cadena de ZooKeeper conexión de Apache. Sin embargo, le recomendamos que utilice `BootstrapServerString` para conectarse a su clúster y realizar operaciones de administración, ya que el indicador `--zookeeper` ha quedado obsoleto en Kafka 2.5 y se ha eliminado de Kafka 3.0.

Obtener la cadena de ZooKeeper conexión de Apache mediante el AWS Management Console

1. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/.
2. La tabla muestra todos los clústeres de la región actual en esta cuenta. Elija el nombre de un clúster para ver su descripción.
3. En la página Resumen del clúster, elija Ver información del cliente. Aquí se muestran los agentes de arranque, así como la cadena de ZooKeeper conexión de Apache.

Obtención de la cadena de ZooKeeper conexión de Apache mediante AWS CLI

1. Si no conocee el nombre de recurso de Amazon (ARN) de su clúster, puede encontrarlo enumerando todos los clústeres de su cuenta. Para obtener más información, consulte the section called "Enumerar clústeres".
2. Para obtener la cadena de ZooKeeper conexión de Apache, junto con otra información sobre el clúster, ejecute el siguiente comando y *`ClusterArn`* reemplácelo por el ARN del clúster.

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

   El resultado de este comando `describe-cluster` tendrá un aspecto similar al siguiente.

   ```
   {
       "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"
    }
}
```

El ejemplo de JSON anterior muestra la clave `ZookeeperConnectString` en la salida del comando `describe-cluster`. Copie el valor correspondiente a esta clave y guárdelo para cuando necesite crear un tema en el clúster.

> **Important**
>
> Su clúster de Amazon MSK debe estar en `ACTIVE` ese estado para que pueda obtener la cadena de ZooKeeper conexión de Apache. Cuando un clúster todavía está en el estado `CREATING`, la salida del comando `describe-cluster` no incluye `ZookeeperConnectString`. Si este es el caso, espere unos minutos y vuelva a ejecutar `describe-cluster` después de que el clúster alcance el estado `ACTIVE`.

Obtener la cadena de ZooKeeper conexión de Apache mediante la API

Para obtener la cadena de ZooKeeper conexión de Apache mediante la API, consulte DescribeCluster.

KRaft modo

Amazon MSK introdujo la compatibilidad con KRaft (Apache Kafka Raft) en la versión 3.7.x de Kafka. La comunidad de Apache Kafka se desarrolló KRaft para reemplazar a Apache en la administración de metadatos en los clústeres de Apache ZooKeeper Kafka. En el KRaft modo, los metadatos del clúster se propagan dentro de un grupo de controladores de Kafka, que forman parte del clúster de Kafka, en lugar de propagarse entre los nodos. ZooKeeper KRaftLos controladores se incluyen sin coste adicional para usted y no requieren ninguna configuración o administración adicionales por su parte. Consulte el KIP-500 para obtener más información sobre. KRaft

Estos son algunos puntos a tener en cuenta sobre el KRaft modo en MSK:

- KRaft el modo solo está disponible para clústeres nuevos. Una vez creado el clúster, no se puede cambiar de modo de metadatos.
- En la consola de MSK, puede crear un clúster basado en Kraft seleccionando la versión 3.7.x de Kafka y marcando la casilla de verificación de la ventana de creación del KRaft clúster.
- Para crear un clúster en KRaft modo mediante la API `CreateCluster`o `CreateClusterV2`las operaciones de MSK, debe utilizarlo como versión. `3.7.x.kraft` `3.7.x`Utilícela como versión para crear un clúster en ZooKeeper modo.
- El número de particiones por agente es el mismo en los clústeres KRaft y ZooKeeper basados en ellos. Sin embargo, KRaft le permite alojar más particiones por clúster al aprovisionar más agentes en un clúster.
- No es necesario realizar cambios en la API para usar el KRaft modo en Amazon MSK. Sin embargo, si sus clientes siguen utilizando la cadena de conexión `--zookeeper` en la actualidad, debe actualizar a sus clientes para que usen la cadena de conexión `--bootstrap-server` para conectarse a su clúster. El indicador `--zookeeper` está obsoleto en la versión 2.5 de Apache Kafka y se elimina a partir de la versión 3.0 de Kafka. Por lo tanto, le recomendamos que utilice las versiones recientes del cliente Apache Kafka y la cadena de conexión `--bootstrap-server` para todas las conexiones a su clúster.
- ZooKeeper El modo sigue estando disponible en todas las versiones publicadas en las que Apache Kafka también admite zookeeper. Consulte Versiones compatibles de Apache Kafka para

obtener más información sobre la finalización de la compatibilidad con las versiones y futuras actualizaciones de Apache Kafka.

- Debe comprobar que todas las herramientas que utilice son capaces de utilizar Kafka Admin sin necesidad de conexiones. APIs ZooKeeper Consulte Utilice LinkedIn el control de crucero para Apache Kafka con Amazon MSK para conocer los pasos actualizados para conectar el clúster a Cruise Control. El Cruise Control también incluye instrucciones para utilizar el Cruise Control sin ZooKeeper él.
- No necesita acceder directamente a los KRaft controladores del clúster para realizar ninguna acción administrativa. Sin embargo, si utiliza la supervisión abierta para recopilar métricas, también necesitará los puntos de conexión de DNS de sus controladores para recopilar algunas métricas del clúster que no estén relacionadas con los controladores. Puede obtener estos puntos finales de DNS desde la consola de MSK o mediante la operación de la ListNodesAPI. Consulte Supervise un clúster aprovisionado de MSK con Prometheus los pasos actualizados para configurar la supervisión abierta de los clústeres basados. KRaft
- No hay CloudWatch métricas adicionales que debas supervisar para los clústeres de KRaft modos sobre los clústeres de ZooKeeper modos. MSK administra los KRaft controladores que se utilizan en los clústeres.
- Puede seguir gestionando el ACLs uso de clústeres en KRaft modo activo mediante la cadena de `--bootstrap-server` conexión. No debe usar la cadena de `--zookeeper` conexión para administrar ACLs. Consulte Apache Kafka ACLs.
- En KRaft el modo, los metadatos del clúster se almacenan en KRaft controladores de Kafka y no en ZooKeeper nodos externos. Por lo tanto, no necesita controlar el acceso a los nodos del controlador por separado, como lo hace con ZooKeeper los nodos.

Recursos de Amazon MSK

El término recursos tiene dos significados en Amazon MSK, según el contexto. En el contexto de APIs un recurso, es una estructura en la que se puede invocar una operación. Para obtener una lista de estos recursos y las operaciones que puede invocar en ellos, consulte Recursos en la referencia de la API de Amazon MSK. En el contexto de the section called “Control de acceso de IAM”, un recurso es una entidad a la que puede permitir o denegar el acceso, tal y como se define en la sección the section called “Recursos de la política de autorización”.

Versiones de Apache Kafka

Cuando se crea un clúster de Amazon MSK, debe especificar la versión de Apache Kafka que desea que tenga. También puede actualizar la versión de Apache Kafka de un clúster existente. Los temas que se presentan en este capítulo lo ayudan a comprender los plazos del soporte de las versiones de Kafka y las prácticas recomendadas.

Temas

- Versiones compatibles de Apache Kafka
- Soporte de las versiones de Amazon MSK

Versiones compatibles de Apache Kafka

Amazon Managed Streaming para Apache Kafka (Amazon MSK) es compatible con las siguientes versiones de Apache Kafka y Amazon MSK. La comunidad de Apache Kafka ofrece soporte para cada versión durante aproximadamente 12 meses después de su fecha de publicación. Para obtener más detalles, consulte la política de EOL (fin de vida útil) de Apache Kafka.

Versiones compatibles de Apache Kafka

Versión de Apache Kafka	Fecha de publicación de MSK	Fecha de finalización del soporte
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	2020-09-09	2024-06-08
2.5.1	2020-09-30	2024-06-08
2.6.0	2020-10-21	2024-09-11
2.6.1	2021-01-19	2024-09-11

Versión de Apache Kafka	Fecha de publicación de MSK	Fecha de finalización del soporte
2.6.2	2021-04-29	2024-09-11
2.6.3	2021-12-21	2024-09-11
2.7.0	2020-12-29	2024-09-11
2.7.1	2021-05-25	2024-09-11
2.7.2	2021-12-21	2024-09-11
2.8.0	2021-05-19	2024-09-11
2.8.1	2022-10-28	2024-09-11
2.8.2, en niveles	28 de octubre de 2022	2025-01-14
3.1.1	2022-06-22	2024-09-11
3.2.0	2022-06-22	2024-09-11
3.3.1	2022-10-26	2024-09-11
3.3.2	2023-03-02	2024-09-11
3.4.0	2023-05-04	2025-08-04
3.5.1	2023-09-26	2025-10-23
3.6.0	2023-11-16	--
3.7.x	2024-05-29	--
3.8.x	2025-02-20	--
3.9.x	2025-04-21	--
4.0.x	2025-05-16	--

Para obtener más información sobre la política de soporte de versiones de Amazon MSK, consulte Política de soporte de las versiones de Amazon MSK.

Amazon MSK versión 4.0.x

Amazon Managed Streaming for Apache Kafka (Amazon MSK) ahora es compatible con la versión 4.0 de Apache Kafka. Esta versión incorpora los últimos avances en la gestión y el rendimiento de los clústeres a MSK Provisioned. Kafka 4.0 presenta un nuevo protocolo de reequilibrio para el consumidor, que ahora está disponible de forma generalizada, que ayuda a garantizar un reequilibrio de grupos más fluido y rápido. Además, Kafka 4.0 requiere que los intermediarios y las herramientas utilicen Java 17, ya que proporciona seguridad y rendimiento mejorados, incluye diversas correcciones de errores y mejoras, y desaprueba la administración de metadatos mediante Apache. ZooKeeper

Para obtener más información y una lista completa de mejoras y correcciones de errores, consulte las notas de publicación de Apache Kafka para la versión 4.0.

Amazon MSK versión 3.9.x

Amazon Managed Streaming for Apache Kafka (Amazon MSK) ahora es compatible con la versión 3.9 de Apache Kafka. Esta versión le permite conservar los datos en niveles al deshabilitar el almacenamiento en niveles a nivel temático. Las aplicaciones de consumo pueden seguir leyendo los datos históricos de la compensación de inicio del registro remoto (Rx) y, al mismo tiempo, mantener las compensaciones de registro continuas en el almacenamiento local y remoto.

Para obtener más información y una lista completa de mejoras y correcciones de errores, consulte las notas de la versión 3.9.x de Apache Kafka.

Amazon MSK versión 3.8.x

Amazon Managed Streaming for Apache Kafka (Amazon MSK) ahora es compatible con la versión 3.8 de Apache Kafka. Ahora puede crear nuevos clústeres con la versión 3.8 con KRAFT o con el ZooKeeper modo de administración de metadatos, o bien actualizar los clústeres ZooKeeper basados existentes para usar la versión 3.8. La versión 3.8 de Apache Kafka incluye varias correcciones de errores y nuevas funciones que mejoran el rendimiento. Las nuevas características clave incluyen la compatibilidad con la configuración del nivel de compresión. Esto le permite optimizar aún más el rendimiento al utilizar tipos de compresión como lz4, zstd y gzip, ya que le permite cambiar el nivel de compresión predeterminado.

Para obtener más información y una lista completa de mejoras y correcciones de errores, consulte las notas de la versión 3.8.x de Apache Kafka.

Versión 3.7.x Apache Kafka (con almacenamiento en niveles listo para producción)

La versión 3.7.x de Apache Kafka en MSK incluye compatibilidad para la versión 3.7.0 de Apache Kafka. Para utilizar la nueva versión 3.7.x, puede crear clústeres o actualizar los existentes. Con este cambio en el nombre de las versiones, ya no tendrá que adoptar versiones más recientes de corrección de parches, como la 3.7.1, cuando las publique la comunidad de Apache Kafka. Amazon MSK actualizará automáticamente la versión 3.7.x para que sea compatible con las futuras versiones de parche una vez que estén disponibles. Esto permite aprovechar las correcciones de seguridad y errores disponibles con las versiones de corrección de parches sin activar una actualización de versión. Estas versiones de corrección de parches que publica Apache Kafka no rompen la compatibilidad con la versión. Además, puede beneficiarse de las nuevas versiones de corrección de parches sin tener que preocuparse por los errores de lectura o escritura en las aplicaciones de sus clientes. Asegúrese de que sus herramientas de automatización de infraestructuras, por ejemplo CloudFormation, estén actualizadas para tener en cuenta este cambio en el nombre de las versiones.

Amazon MSK ahora admite el KRaft modo (Apache Kafka Raft) en la versión 3.7.x de Apache Kafka. En Amazon MSK, al igual que con ZooKeeper los nodos, los KRaft controladores se incluyen sin coste adicional para usted y no requieren ninguna configuración o administración adicionales por su parte. Ahora puede crear clústeres en cualquier KRaft modo o en modo en la versión ZooKeeper 3.7.x de Apache Kafka. En el modo KRaft, puede agregar hasta 60 agentes y alojar más particiones por clúster, sin que se requiera un aumento en el límite, en comparación con la cuota de 30 agentes de los clústeres basados en ZooKeeper. Para obtener más información KRaft sobre MSK, consulte. KRaft modo

La versión 3.7.x de Apache Kafka también incluye varias correcciones de errores y nuevas características que mejoran el rendimiento. Entre las mejoras más importantes, se incluyen las optimizaciones de detección de nodos principales para los clientes y las opciones de optimización del vaciado de los segmentos de registros. Para consultar la lista completa de mejoras y correcciones de errores, consulte las notas de la versión 3.7.0 de Apache Kafka.

Versión 3.6.0 de Apache Kafka (con almacenamiento en niveles listo para producción)

Para obtener información sobre la versión 3.6.0 de Apache Kafka (con almacenamiento en niveles listo para producción), consulte las notas de la versión en el sitio de descargas de Apache Kafka.

Amazon MSK seguirá utilizando y gestionando Zookeeper para la gestión del cuórum en esta versión para garantizar la estabilidad.

Versión 3.5.1 de Amazon MSK

Amazon Managed Streaming para Apache Kafka (Amazon MSK) ahora es compatible con la versión 3.5.1 de Apache Kafka para clústeres nuevos y existentes. La versión 3.5.1 de Apache Kafka incluye varias correcciones de errores y nuevas características que mejoran el rendimiento. Entre las características principales, se incluye la introducción de una nueva asignación de particiones compatible con los bastidores para los consumidores. Amazon MSK seguirá utilizando y administrando Zookeeper para la administración del cuórum en esta versión. Para consultar la lista completa de mejoras y correcciones de errores, consulte las notas de la versión 3.5.1 de Apache Kafka.

Para obtener información sobre la versión 3.5.1 de Apache Kafka, consulte las notas de la versión en el sitio de descargas de Apache Kafka.

Versión 3.4.0 de Amazon MSK

Amazon Managed Streaming para Apache Kafka (Amazon MSK) ahora es compatible con la versión 3.4.0 de Apache Kafka para clústeres nuevos y existentes. La versión 3.4.0 de Apache Kafka incluye varias correcciones de errores y nuevas características que mejoran el rendimiento. Entre las características principales, se incluye una corrección para mejorar la estabilidad al recuperar datos de la réplica más cercana. Amazon MSK seguirá utilizando y administrando Zookeeper para la administración del cuórum en esta versión. Para consultar la lista completa de mejoras y correcciones de errores, consulte las notas de la versión 3.4.0 de Apache Kafka.

Para obtener información sobre la versión 3.4.0 de Apache Kafka, consulte las notas de la versión en el sitio de descargas de Apache Kafka.

Versión 3.3.2 de Amazon MSK

Amazon Managed Streaming para Apache Kafka (Amazon MSK) ahora es compatible con la versión 3.3.2 de Apache Kafka para clústeres nuevos y existentes. La versión 3.3.2 de Apache Kafka incluye varias correcciones de errores y nuevas características que mejoran el rendimiento. Entre las características principales, se incluye una corrección para mejorar la estabilidad al recuperar datos de la réplica más cercana. Amazon MSK seguirá utilizando y administrando Zookeeper para la administración del cuórum en esta versión. Para consultar la lista completa de mejoras y correcciones de errores, consulte las notas de la versión 3.3.2 de Apache Kafka.

Para obtener información sobre la versión 3.3.2 de Apache Kafka, consulte las notas de la versión en el sitio de descargas de Apache Kafka.

Versión 3.3.1 de Amazon MSK

Amazon Managed Streaming para Apache Kafka (Amazon MSK) ahora es compatible con la versión 3.3.1 de Apache Kafka para clústeres nuevos y existentes. La versión 3.3.1 de Apache Kafka incluye varias correcciones de errores y nuevas características que mejoran el rendimiento. Entre las características más importantes, se incluyen mejoras en las métricas y en el particionador. Amazon MSK seguirá utilizando y gestionando Zookeeper para la gestión del cuórum en esta versión para garantizar la estabilidad. Para consultar la lista completa de mejoras y correcciones de errores, consulte las notas de la versión 3.3.1 de Apache Kafka.

Para obtener información sobre la versión 3.3.1 de Apache Kafka, consulte las notas de la versión en el sitio de descargas de Apache Kafka.

Versión 3.1.1 de Amazon MSK

Amazon Managed Streaming para Apache Kafka (Amazon MSK) ahora es compatible con las versiones 3.1.1 y 3.2.0 de Apache Kafka para clústeres nuevos y existentes. Las versiones 3.1.1 y 3.2.0 de Apache Kafka incluyen varias correcciones de errores y nuevas características que mejoran el rendimiento. Algunas de las características clave incluyen mejoras en las métricas y el uso del tema IDs. MSK seguirá utilizando y administrando Zookeeper para la administración del cuórum en esta versión para garantizar la estabilidad. Para consultar la lista completa de mejoras y correcciones de errores, consulte las notas de las versiones 3.1.1 y 3.2.0 de Apache Kafka.

Para obtener información sobre las versiones 3.1.1 y 3.2.0 de Apache Kafka, consulte las notas de la versión 3.2.0 y las notas de la versión 3.1.1 en el sitio de descargas de Apache Kafka.

Almacenamiento por niveles de Amazon MSK, versión 2.8.2

Esta versión es una versión exclusiva para Amazon MSK de la versión 2.8.2 de Apache Kafka y es compatible con los clientes Apache Kafka de código abierto.

La versión 2.8.2 contiene una funcionalidad de almacenamiento por niveles que es compatible con la APIs introducida en el KIP-405 para Apache Kafka. Para obtener más información acerca de la característica por niveles de Amazon MSK, consulte Almacenamiento por niveles para corredores estándar.

Versión 2.5.1 de Apache Kafka

La versión 2.5.1 de Apache Kafka incluye varias correcciones de errores y nuevas funciones, como el cifrado en tránsito para los clientes de administración y de Apache. ZooKeeper Amazon

MSK proporciona ZooKeeper puntos de enlace TLS, que puede consultar con la operación. DescribeCluster

El resultado de la DescribeClusteroperación incluye el `ZookeeperConnectStringTls` nodo, que muestra los puntos de enlace de TLS Zookeeper.

El siguiente ejemplo muestra el nodo `ZookeeperConnectStringTls` de la respuesta de la operación `DescribeCluster`:

```
"ZookeeperConnectStringTls": "z-3.awskafkatutorialc.abcd123.c3.kafka.us-
east-1.amazonaws.com:2182,z-2.awskafkatutorialc.abcd123.c3.kafka.us-
east-1.amazonaws.com:2182,z-1.awskafkatutorialc.abcd123.c3.kafka.us-
east-1.amazonaws.com:2182"
```

Para obtener información sobre el uso del cifrado TLS con Zookeeper, consulte Uso de la seguridad TLS con Apache ZooKeeper.

Para obtener más información sobre la versión 2.5.1 de Apache Kafka, consulte las notas de la versión en el sitio de descargas de Apache Kafka.

Solución de errores de Amazon MSK, versión 2.4.1.1

Esta versión es una versión de corrección de errores exclusiva para Amazon MSK de la versión 2.4.1 de Apache Kafka. Esta versión de corrección de errores contiene una solución para el KAFKA-9752, un problema poco frecuente que provoca que los grupos de consumidores se reequilibren continuamente y permanezcan en el estado `PreparingRebalance`. Este problema afecta a los clústeres que ejecutan las versiones 2.3.1 y 2.4.1 de Apache Kafka. Esta versión contiene una corrección producida por la comunidad que está disponible en la versión 2.5.0 de Apache Kafka.

> **Note**
>
> Los clústeres de Amazon MSK que ejecutan la versión 2.4.1.1 son compatibles con cualquier cliente de Apache Kafka que sea compatible con la versión 2.4.1 de Apache Kafka.

Si prefiere usar Apache Kafka 2.4.1, le recomendamos que utilice la versión 2.4.1.1 con la corrección de errores de MSK para clústeres nuevos de Amazon MSK. Puede actualizar los clústeres existentes que ejecutan la versión 2.4.1 de Apache Kafka a esta versión para incorporar esta corrección. Para

obtener información sobre cómo actualizar un clúster existente, consulte Actualice la versión de Apache Kafka.

Para solucionar este problema sin actualizar el clúster a la versión 2.4.1.1, consulte la sección Grupo de consumidores atrapado en el estado `PreparingRebalance` de la guía Solución de problemas del clúster de Amazon MSK.

Versión 2.4.1 de Apache Kafka (utilice 2.4.1.1 en su lugar)

> **Note**
>
> Ya no puede crear un clúster de MSK con la versión 2.4.1 de Apache Kafka. En su lugar, puede usar Solución de errores de Amazon MSK, versión 2.4.1.1 con clientes compatibles con la versión 2.4.1 de Apache Kafka. Y si ya tiene un clúster de MSK con la versión 2.4.1 de Apache Kafka, le recomendamos que lo actualice para que utilice la versión 2.4.1.1 de Apache Kafka en su lugar.

KIP-392 es una de las principales propuestas de mejora de Kafka que se incluyen en la versión 2.4.1 de Apache Kafka. Esta mejora permite a los consumidores recuperar de la réplica más cercana. Para utilizar esta característica, establezca `client.rack` en las propiedades del consumidor en el ID de la zona de disponibilidad del consumidor. Un ejemplo de ID de AZ es `use1-az1`. Amazon MSK establece `broker.rack` las zonas IDs de disponibilidad de los corredores. También debe establecer la propiedad de configuración `replica.selector.class` en `org.apache.kafka.common.replica.RackAwareReplicaSelector`, que es una implementación de reconocimiento de bastidor proporcionada por Apache Kafka.

Cuando utiliza esta versión de Apache Kafka, las métricas en el nivel de monitoreo abierto `PER_TOPIC_PER_BROKER` aparecen solo después de que sus valores sean distintos de cero por primera vez. Para obtener más información acerca de este tema, consulte the section called “Supervisión de `PER_TOPIC_PER_BROKER`”.

Para obtener información sobre cómo encontrar la zona de disponibilidad IDs, consulte AZ IDs for Your Resource en la guía del AWS Resource Access Manager usuario.

Para obtener información sobre los ajustes de las propiedades de configuración, consulte the section called “Configuración de bróker”.

Para obtener más información acerca de KIP-392, consulte Allow Consumers to Fetch from Closest Replica en las páginas de Confluence.

Para obtener más información sobre la versión 2.4.1 de Apache Kafka, consulte las notas de la versión en el sitio de descargas de Apache Kafka.

Soporte de las versiones de Amazon MSK

Este tema describe la Política de soporte de las versiones de Amazon MSK y el procedimiento de Actualice la versión de Apache Kafka. Si actualiza desde su versión de Kafka, siga las prácticas recomendadas que se describen en Prácticas recomendadas para la actualización de versiones.

Temas

- Política de soporte de las versiones de Amazon MSK
- Actualice la versión de Apache Kafka
- Prácticas recomendadas para la actualización de versiones

Política de soporte de las versiones de Amazon MSK

En esta sección, se describe la política de soporte para las versiones de Kafka compatibles con Amazon MSK.

- Todas las versiones de Kafka son compatibles hasta la fecha de finalización del soporte. Para obtener información sobre las fechas de finalización del soporte, consulte Versiones compatibles de Apache Kafka. Actualice su clúster MSK a la versión recomendada de Kafka o a una superior antes de la fecha de finalización del soporte. Para obtener más información sobre cómo actualizar su versión de Apache Kafka, consulteActualice la versión de Apache Kafka. Cualquier clúster que utilice una versión de Kafka posterior a la fecha de fin del soporte se actualizará automáticamente a la versión recomendada de Kafka. Las actualizaciones automáticas se pueden realizar en cualquier momento después de la fecha de finalización del soporte. No recibirá ninguna notificación antes de la actualización.
- MSK eliminará gradualmente el soporte para los clústeres recién creados que utilicen versiones de Kafka con fechas de fin de soporte publicadas.

Actualice la versión de Apache Kafka

Puede actualizar un clúster de MSK existente a una versión más reciente de Apache Kafka.

> **Note**
>
> - No puede actualizar un clúster de MSK existente de una versión ZooKeeper basada en Apache Kafka a una versión más reciente que utilice o requiera el modo. KRaft En su lugar, para actualizar el clúster, cree un nuevo clúster de MSK con una versión de Kafka KRaft compatible y migre los datos y las cargas de trabajo del clúster anterior.
> - Amazon MSK solo actualiza el software del servidor. No actualiza a sus clientes.
> - No puede degradar un clúster de MSK existente a una versión anterior de Apache Kafka.

Cuando actualice la versión de Apache Kafka de un clúster de MSK, compruebe también el software del lado del cliente para asegurarse de que esa versión le permite utilizar las funciones de la nueva versión de Apache Kafka del clúster.

Para obtener información sobre cómo hacer que un clúster esté altamente disponible durante una actualización, consulte. the section called “Crear clústeres de alta disponibilidad”

Actualice la versión de Apache Kafka mediante el AWS Management Console

1. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/.
2. En la barra de navegación, elija la región en la que creó el clúster de MSK.
3. Elija el clúster de MSK que desee actualizar.
4. En la pestaña Propiedades, elija Actualizar en la sección de versiones de Apache Kafka.
5. En la sección de la versión de Apache Kafka, haga lo siguiente:

    a. En la lista desplegable Elija la versión de Apache Kafka, elija la versión a la que desee actualizar. En este ejemplo, elija **`3.9.x`**.

    b. (Opcional) Seleccione Compatibilidad de versiones para ver la compatibilidad entre la versión actual del clúster y la versión a la que desea actualizar. A continuación, elija Elegir continuar o Cancelar.

    c. Seleccione la casilla Actualizar la configuración del clúster para aplicar automáticamente una nueva revisión de la configuración de Kafka que sea compatible con la versión actualizada. Esto garantiza la compatibilidad y permite añadir nuevas funciones o mejoras a la versión actualizada. Sin embargo, omítala si quieres mantener tus configuraciones personalizadas existentes.

    d. Seleccione Actualizar.

Actualice la versión de Apache Kafka mediante el AWS CLI

1. Ejecute el siguiente comando, reemplazando *ClusterArn* por el nombre de recurso de Amazon (ARN) que obtuvo al crear el clúster. Si no tiene el ARN para su clúster, puede encontrarlo enumerando todos los clústeres. Para obtener más información, consulte the section called "Enumerar clústeres".

   ```
   aws kafka get-compatible-kafka-versions --cluster-arn ClusterArn
   ```

   El resultado de este comando incluye una lista de las versiones de Apache Kafka a las que puede actualizar el clúster. Es similar al ejemplo siguiente.

   ```
   {
       "CompatibleKafkaVersions": [
           {
               "SourceVersion": "2.2.1",
               "TargetVersions": [
                   "2.3.1",
                   "2.4.1",
                   "2.4.1.1",
                   "2.5.1"
               ]
           }
       ]
   }
   ```

2. Ejecute el siguiente comando, reemplazando *ClusterArn* por el nombre de recurso de Amazon (ARN) que obtuvo al crear el clúster. Si no tiene el ARN para su clúster, puede encontrarlo enumerando todos los clústeres. Para obtener más información, consulte the section called "Enumerar clústeres".

   Reemplace *Current-Cluster-Version* con la versión actual del clúster. Pues *TargetVersion* puede especificar cualquiera de las versiones de destino a partir del resultado del comando anterior.

> **Important**
>
> Las versiones de clúster no son enteros simples. Para encontrar la versión actual del clúster, utilice la DescribeClusteroperación o el comando AWS CLI describe-cluster. Un ejemplo de ID de versión es `KTVPDKIKX0DER`.

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

El resultado del comando anterior tiene un aspecto similar al siguiente JSON.

```
{

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

3. Para obtener el resultado de la `update-cluster-kafka-version` operación, ejecute el siguiente comando y *ClusterOperationArn* reemplácelo por el ARN que obtuvo en el resultado del `update-cluster-kafka-version` comando.

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

El resultado de este comando `describe-cluster-operation` tendrá un aspecto similar al siguiente.

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

Si `OperationState` tiene el valor `UPDATE_IN_PROGRESS`, espere un rato y vuelva a ejecutar el comando `describe-cluster-operation`. Cuando se completa la operación, el valor de `OperationState` se convierte en `UPDATE_COMPLETE`. Como el tiempo necesario para que Amazon MSK complete la operación varía, es posible que tenga que comprobarlo varias veces hasta que se complete la operación.

Actualice la versión de Apache Kafka mediante la API

1. Ejecute la GetCompatibleKafkaVersionsoperación para obtener una lista de las versiones de Apache Kafka a las que puede actualizar el clúster.

2. Ejecute la UpdateClusterKafkaVersionoperación para actualizar el clúster a una de las versiones compatibles de Apache Kafka.

Prácticas recomendadas para la actualización de versiones

Para garantizar la continuidad del cliente durante la actualización progresiva que se realiza como parte del proceso de actualización de la versión de Kafka, revise la configuración de sus clientes y los temas de Apache Kafka de la siguiente manera:

- Establezca el factor de replicación (RF) del tema en un valor mínimo de 2 para clústeres de dos zonas de disponibilidad y un valor mínimo de 3 para los clústeres de tres zonas de disponibilidad. Si el valor de RF es 2, se pueden producir particiones fuera de línea durante la aplicación del parche.
- Establezca el mínimo de réplicas sincronizadas (miniSR) en un valor máximo de 1 menos que su factor de replicación (RF), que es. `miniISR = (RF) - 1` Esto garantiza que el conjunto de réplicas de particiones pueda tolerar que una réplica esté fuera de línea o no se replique lo suficiente.
- Configure los clientes para que utilicen varias cadenas de conexión de agentes. Tener varios corredores en la cadena de conexión de un cliente permite la conmutación por error si un corredor específico que brinda soporte al cliente I/O comienza a recibir parches. Para obtener información sobre cómo obtener una cadena de conexión con varios agentes, consulte Obtención de agentes de arranque para un clúster de Amazon MSK.
- Recomendamos que actualice los clientes conectados a la versión recomendada o superior para aprovechar las características disponibles en la nueva versión. Las actualizaciones de los clientes no están sujetas a las fechas de fin de vida útil (EOL) de la versión Kafka del clúster de MSK y no es necesario que se completen antes de la fecha de EOL. Apache Kafka ofrece una política de compatibilidad bidireccional de clientes que permite que los clientes antiguos trabajen con clústeres más nuevos y viceversa.
- Es probable que los clientes de Kafka que utilizan las versiones 3.x.x tengan los siguientes valores predeterminados: `acks=all` y `enable.idempotence=true`. El valor `acks=all` difiere del valor predeterminado anterior de `acks=1` y ofrece durabilidad adicional al garantizar que las réplicas sincronizadas reconozcan la solicitud de producción. Del mismo modo, el valor predeterminado `enable.idempotence` era anteriormente `false`. El cambio a `enable.idempotence=true` como el valor predeterminado disminuye la posibilidad de mensajes duplicados. Estos cambios se consideran configuraciones de prácticas recomendadas y pueden introducir una pequeña cantidad de latencia adicional que se encuentra dentro de los parámetros normales de rendimiento.

- Al crear nuevos clústeres de MSK, utilice la versión recomendada de Kafka. Si utiliza la versión recomendada de Kafka, podrá beneficiarse de las últimas características de Kafka y MSK.

Solución de problemas del clúster de Amazon MSK

La siguiente información le puede ser de ayuda para solucionar los problemas que podrían presentarse con el clúster de Amazon MSK. También puede publicar el problema en AWS re:Post. Para solucionar problemas del Replicador Amazon MSK, consulte Solución de problemas del Replicador MSK.

Temas

- La sustitución del volumen provoca la saturación del disco debido a la sobrecarga de replicación
- Grupo de consumidores atrapado en el estado PreparingRebalance
- Error al entregar los registros de los corredores a Amazon CloudWatch Logs
- Ningún grupo de seguridad predeterminado
- El clúster aparece atascado en el estado CREATING (Creando)
- El estado del clúster pasa de CREATING (Creando) a FAILED (Error)
- El estado del clúster es ACTIVE (Activo), pero los productores no pueden enviar datos o los consumidores no pueden recibir datos
- AWS CLI no reconoce Amazon MSK
- Las particiones se desconectan o las réplicas no están sincronizadas
- El espacio en el disco se está agotando
- La memoria se está agotando
- El productor obtiene NotLeaderForPartitionException
- Particiones subreplicadas (URP) mayores que cero
- El clúster tiene temas denominados __amazon_msk_canary y __amazon_msk_canary_state
- La replicación de la partición falla
- No se puede acceder al clúster que tiene activado el acceso público
- No se puede acceder al clúster desde dentro AWS: problemas de red
- Error en la autenticación: demasiadas conexiones
- Autenticación fallida: sesión demasiado corta
- MSK sin servidor: se produce un error al crear el clúster

- No se puede actualizar KafkaVersionsList en la configuración de MSK

La sustitución del volumen provoca la saturación del disco debido a la sobrecarga de replicación

Si se produce un error imprevisto en el hardware de volumen, Amazon MSK puede sustituir el volumen con una nueva instancia. Kafka replica las particiones de otros agentes del clúster para rellenar el nuevo volumen. Una vez que las particiones se replican y se actualizan, son elegibles para obtener la pertenencia como réplicas principales y sincronizadas (ISR).

Problema

En un agente que se está recuperando de una sustitución de volumen, algunas particiones de distintos tamaños pueden volver a funcionar antes que otras. Esto puede causar problemas, ya que esas particiones pueden recibir tráfico del mismo agente que sigue actualizando (replicando) otras particiones. Este tráfico de replicación suele saturar los límites de rendimiento del volumen subyacente, que en el caso predeterminado son de 250 MiB por segundo. Cuando se produce esta saturación, las particiones que ya estén actualizadas se verán afectadas, lo que provocará una latencia en todo el clúster para cualquier agente que comparta el ISR con esas particiones actualizadas (no solo para las particiones principales, debido a los reconocimientos `acks=all` remotos). Es más común encontrar esto en los clústeres más grandes, con un mayor número de particiones de diferentes tamaños.

Recomendación

- Para mejorar la estrategia I/O de replicación, asegúrese de contar con la configuración de subprocesos recomendada.
- Para reducir la probabilidad de que se produzca una saturación del volumen subyacente, habilite el almacenamiento aprovisionado con un mayor rendimiento. Se MiB/s recomienda un valor de rendimiento mínimo de 500 para los casos de replicación de alto rendimiento, pero el valor real necesario variará según el rendimiento y el caso de uso. Aprovisione el rendimiento del almacenamiento para los corredores estándar en un clúster de Amazon MSK.
- Para minimizar la presión de replicación, baje `num.replica.fetchers` hasta el valor predeterminado de 2.

Grupo de consumidores atrapado en el estado `PreparingRebalance`

Si uno o más de sus grupos de consumidores están atrapados en un estado de reequilibrio continuo, la causa podría ser el problema KAFKA-9752, que afecta a las versiones 2.3.1 y 2.4.1 de Apache Kafka.

Para resolver este problema, le recomendamos que actualice el clúster a Solución de errores de Amazon MSK, versión 2.4.1.1, que contiene una solución para este problema. Para obtener información sobre cómo actualizar un clúster existente a la versión 2.4.1.1 de corrección de errores de Amazon MSK, consulte Actualice la versión de Apache Kafka.

Las soluciones alternativas para resolver este problema sin actualizar el clúster a la versión 2.4.1.1 de corrección de errores de Amazon MSK consisten en configurar los clientes de Kafka que van a utilizar Protocolo de pertenencia estática, o bien Identificación y reinicio el nodo del agente coordinador del grupo de consumidores atascados.

Implementación de un protocolo de pertenencia estática

Para implementar el protocolo de permanencia estática en los clientes, haga lo siguiente:

1. Establezca la propiedad `group.instance.id` de la configuración de los consumidores de Kafka en una cadena estática que identifique al consumidor del grupo.
2. Asegúrese de que las demás instancias de la configuración estén actualizadas para usar la cadena estática.
3. Implemente los cambios en los consumidores de Kafka.

El uso del protocolo de permanencia estática es más eficaz si el tiempo de espera de la sesión en la configuración del cliente se establece en una duración que permita al consumidor recuperarse sin provocar un reequilibrio prematuro del grupo de consumidores. Por ejemplo, si su aplicación de consumo puede tolerar 5 minutos de inactividad, un valor razonable para el tiempo de espera de la sesión sería de 4 minutos en lugar del valor predeterminado de 10 segundos.

> **Note**
>
> El uso del protocolo de permanencia estática solo reduce la probabilidad de que se produzca este problema. Es posible que siga encontrándose con este problema incluso cuando utilice el protocolo de permanencia estática.

Reinicio del nodo del agente coordinador

Para reiniciar el nodo del agente coordinador, haga lo siguiente:

1. Identifique al coordinador del grupo mediante el comando `kafka-consumer-groups.sh`.
2. Reinicie el coordinador de grupo del grupo de consumidores atascado mediante la acción de la RebootBrokerAPI.

Error al entregar los registros de los corredores a Amazon CloudWatch Logs

Al intentar configurar el clúster para enviar los registros de los corredores a Amazon CloudWatch Logs, es posible que se produzca una de estas dos excepciones.

Si obtiene una excepción `InvalidInput.LengthOfCloudWatchResourcePolicyLimitExceeded`, vuelva a intentarlo pero utilice grupos de registro que comiencen por `/aws/vendedlogs/`. Para obtener más información, consulte Habilitar el registro desde determinados servicios de Amazon Web Services.

Si recibes una `InvalidInput.NumberOfCloudWatchResourcePoliciesLimitExceeded` excepción, elige una política de Amazon CloudWatch Logs existente en tu cuenta y añádele el siguiente JSON.

```
{"Sid":"AWSLogDeliveryWrite","Effect":"Allow","Principal":
{"Service":"delivery.logs.amazonaws.com"},"Action":
["logs:CreateLogStream","logs:PutLogEvents"],"Resource":["*"]}
```

Si intentas añadir el JSON anterior a una política existente pero aparece un error que indica que has alcanzado la longitud máxima de la política que has elegido, intenta añadir el JSON a otra de tus políticas de Amazon CloudWatch Logs. Tras añadir el JSON a una política existente, intenta configurar de nuevo la entrega de registros de agentes a Amazon Logs. CloudWatch

Ningún grupo de seguridad predeterminado

Si intenta crear un clúster y obtiene un error que indica que no hay ningún grupo de seguridad predeterminado, puede deberse a que está utilizando una VPC compartida con usted. Pídale al administrador que le conceda permiso para describir los grupos de seguridad de esta VPC e inténtelo de nuevo. Para ver un ejemplo de una política que permite esta acción, consulte Amazon EC2: permite gestionar grupos de EC2 seguridad asociados a una VPC específica, mediante programación y en la consola.

El clúster aparece atascado en el estado CREATING (Creando)

A veces, la creación de clústeres puede tardar hasta 30 minutos. Espere 30 minutos y compruebe de nuevo el estado del clúster.

El estado del clúster pasa de CREATING (Creando) a FAILED (Error)

Intente crear el clúster de nuevo.

El estado del clúster es ACTIVE (Activo), pero los productores no pueden enviar datos o los consumidores no pueden recibir datos

- Si la creación del clúster se realiza correctamente (el estado del clúster es `ACTIVE`), pero no puede enviar ni recibir datos, asegúrese de que las aplicaciones de productor y consumidor tengan acceso al clúster. Para obtener más información, consulte la guía en the section called “Creación de un equipo cliente”.

- Si sus productores y consumidores tienen acceso al clúster pero siguen teniendo problemas para producir y consumir datos, la causa podría ser KAFKA-7697, que afecta a Apache Kafka versión 2.1.0 y puede conducir a un punto muerto en uno o más corredores. Considere migrar a Apache Kafka 2.2.1, que no se ve afectado por este error. Para obtener información acerca de cómo efectuar la migración, consulte the section called “Migración a un clúster de Amazon MSK”.

AWS CLI no reconoce Amazon MSK

Si lo tienes AWS CLI instalado, pero no reconoce los comandos de Amazon MSK, actualiza tu versión AWS CLI a la última. Para obtener instrucciones detalladas sobre cómo actualizar el AWS CLI, consulte Instalación del AWS Command Line Interface. Para obtener información sobre cómo utilizar los comandos AWS CLI para ejecutar los comandos de Amazon MSK, consultethe section called “Características y conceptos clave”.

Las particiones se desconectan o las réplicas no están sincronizadas

Estos pueden ser síntomas de poco espacio en el disco. Consulte the section called “El espacio en el disco se está agotando”.

El espacio en el disco se está agotando

Consulte las siguientes prácticas recomendadas para administrar el espacio en disco: the section called “Monitorear el espacio en disco” y the section called “Ajuste los parámetros de retención de datos”.

La memoria se está agotando

Si ve que la métrica `MemoryUsed` está ejecutándose alta o `MemoryFree` está ejecutándose en baja, eso no significa que haya un problema. Apache Kafka está diseñado para usar tanta memoria como sea posible, y lo gestiona de manera óptima.

El productor obtiene NotLeaderForPartitionException

Este suele ser un error transitorio. Establezca el parámetro de configuración `retries` del productor en un valor superior a su valor actual.

Particiones subreplicadas (URP) mayores que cero

Es importante supervisar la métrica `UnderReplicatedPartitions`. En un clúster MSK correcto, esta métrica tiene el valor 0. Si es mayor que cero, podría deberse a una de las siguientes razones.

- Si `UnderReplicatedPartitions` tiene picos, el problema puede ser que el clúster no se aprovisiona con el tamaño correcto para manejar el tráfico entrante y saliente. Consulte the section called “Prácticas recomendadas para agentes”.
- Si siempre `UnderReplicatedPartitions` es superior a 0, incluso durante los períodos de poco tráfico, el problema podría deberse a que has establecido restricciones ACLs que no permiten el acceso a los temas a los corredores. Para replicar particiones, los agentes deben estar autorizados a LEER y DESCRIBIR temas. DESCRIBE (Describir) se concede de forma predeterminada con la autorización READ (Leer). Para obtener información sobre la configuración ACLs, consulte Autorización y la ACLs documentación de Apache Kafka.

El clúster tiene temas denominados __amazon_msk_canary y __amazon_msk_canary_state

Es posible que vea que su clúster de MSK tiene un tema con el nombre `__amazon_msk_canary` y otro con el nombre `__amazon_msk_canary_state`. Se trata de temas internos que Amazon MSK crea y utiliza para las métricas de estado y diagnóstico del clúster. Estos temas tienen un tamaño insignificante y no se pueden eliminar.

La replicación de la partición falla

Asegúrese de no haber configurado ACLs CLUSTER_ACTIONS.

No se puede acceder al clúster que tiene activado el acceso público

Si su clúster tiene activado el acceso público, pero sigue sin poder acceder a él desde Internet, siga estos pasos:

1. Asegúrese de que las reglas de entrada del grupo de seguridad del clúster admitan su dirección IP y el puerto del clúster. Para obtener una lista de los números de puerto del clúster, consulte the section called “Información del puerto”. Asegúrese también de que las reglas de salida del grupo de seguridad permitan las comunicaciones salientes. Para obtener más información acerca del uso de grupos de seguridad de VPC y sus reglas de entrada y salida, consulte Grupos de seguridad de la VPC en la Guía del usuario de Amazon VPC.
2. Asegúrese de que su dirección IP y el puerto del clúster estén permitidos en las reglas de entrada de la ACL de la red de VPC del clúster. A diferencia de los grupos de seguridad, las redes ACLs no tienen estado. Esto significa que debe configurar las reglas de entrada y salida. En las reglas de salida, permita que todo el tráfico (rango de puertos: 0-65535) llegue a su dirección IP. Para obtener más información, consulte Adición y eliminación de reglas en la Guía del usuario de Amazon VPC.
3. Asegúrese de utilizar la cadena bootstrap-brokers de acceso público para acceder al clúster. Un clúster de MSK que tiene activado el acceso público tiene dos cadenas bootstrap-brokers diferentes, una para el acceso público y otra para el acceso desde dentro de AWS. Para obtener más información, consulte the section called “Obtenga los corredores de bootstrap utilizando la AWS Management Console”.

No se puede acceder al clúster desde dentro AWS: problemas de red

Si tiene una aplicación de Apache Kafka que no puede comunicarse correctamente con un clúster de MSK, comience por llevar a cabo la siguiente prueba de conectividad.

1. Utilice cualquiera de los métodos descritos en the section called “Obtenga los corredores de bootstrap” para obtener las direcciones de los agentes de arranque.
2. En el siguiente comando, *`bootstrap-broker`* sustitúyala por una de las direcciones de intermediario que obtuviste en el paso anterior. *`port-number`*Sustitúyala por 9094 si el clúster está configurado para usar la autenticación TLS. Si el clúster no usa la autenticación TLS, *`port-number`* sustitúyala por la 9092. Ejecute el comando desde el equipo cliente.

```
telnet bootstrap-broker port-number
```

Donde port-number es:

- 9094 si el clúster está configurado para utilizar la autenticación TLS.
- 9092 si el clúster no utiliza la autenticación TLS.
- Si el acceso público está habilitado, se requiere un port-number diferente.

Ejecute el comando desde el equipo cliente.

3. Repita el comando anterior para todos los agentes de arranque.

Si el equipo cliente puede acceder a los agentes, esto significa que no hay problemas de conectividad. En este caso, ejecute el siguiente comando para comprobar si su cliente Apache Kafka está configurado correctamente. Para obtenerla*bootstrap-brokers*, utilice cualquiera de los métodos descritos en. the section called “Obtenga los corredores de bootstrap” *topic*Sustitúyalo por el nombre del tema.

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

Si el comando anterior tiene éxito, significa que su cliente está configurado correctamente. Si sigue sin poder producir y consumir desde una aplicación, depure el problema en el nivel de aplicación.

Si el equipo cliente no puede acceder a los agentes, consulte las siguientes subsecciones para obtener instrucciones según la configuración del equipo cliente.

EC2 Cliente de Amazon y clúster de MSK en la misma VPC

Si el equipo cliente está en la misma VPC que el clúster de MSK, asegúrese de que el grupo de seguridad del clúster tenga una regla de entrada que acepte el tráfico del grupo de seguridad del equipo cliente. Para obtener información acerca de estas reglas, consulte Reglas del grupo de seguridad. Para ver un ejemplo de cómo acceder a un clúster desde una EC2 instancia de Amazon que se encuentra en la misma VPC que el clúster, consulta. the section called “Introducción”

EC2 Cliente de Amazon y clúster de MSK en diferentes VPCs

Si la máquina cliente y el clúster están en dos equipos diferentes VPCs, asegúrese de lo siguiente:

- Los dos VPCs están emparejados.
- El estado de la interconexión está activo.
- Las tablas de enrutamiento de los dos VPCs están configuradas correctamente.

Para obtener información acerca de la interconexión de VPC, consulte Trabajo con interconexiones de VPC.

Cliente en las instalaciones

En el caso de un cliente local que esté configurado para conectarse al clúster de MSK mediante AWS VPN, asegúrese de lo siguiente:

- El estado de la conexión de VPN es UP. Para obtener información acerca de cómo comprobar el estado de la conexión de VPN, consulte ¿Cómo compruebo el estado actual de mi túnel VPN?.
- La tabla de enrutamiento de la VPC del clúster contiene la ruta de un CIDR en las instalaciones cuyo destino tiene el formato `Virtual private gateway(vgw-xxxxxxxx)`.
- El grupo de seguridad del clúster de MSK permite el tráfico en el puerto 2181, el puerto 9092 (si el clúster acepta tráfico de texto sin formato) y el puerto 9094 (si el clúster acepta tráfico cifrado TLS).

Para obtener más instrucciones AWS VPN de solución de problemas, consulte Solución de problemas de Client VPN.

AWS Direct Connect

Si el cliente la usa AWS Direct Connect, consulte Solución de problemas AWS Direct Connect.

Si las instrucciones de solución de problemas anteriores no resuelven el problema, asegúrese de que ningún firewall bloquee el tráfico de red. Para depurar más, utilice herramientas como `tcpdump` y `Wireshark` para analizar el tráfico y para asegurarse de que está llegando al clúster de MSK.

Error en la autenticación: demasiadas conexiones

El error `Failed authentication ... Too many connects` indica que un agente se está protegiendo a sí mismo porque uno o varios clientes de IAM están intentando conectarse a él a un ritmo agresivo. Para ayudar a los agentes a aceptar una tasa más alta de nuevas conexiones de IAM, puede aumentar el parámetro de la configuración `reconnect.backoff.ms`.

Para obtener más información sobre los límites de velocidad para las nuevas conexiones por agente, consulte la página Cuota de Amazon MSK.

Autenticación fallida: sesión demasiado corta

El `Failed authentication ... Session too short` error se produce cuando el cliente intenta conectarse a un clúster con credenciales de IAM que están a punto de caducar. Asegúrese de comprobar cómo se actualizan sus credenciales de IAM. Lo más probable es que las credenciales se sustituyan demasiado cerca de la fecha de caducidad de la sesión, lo que provoca problemas en el servidor y fallos de autenticación.

MSK sin servidor: se produce un error al crear el clúster

Si intenta crear un clúster de MSK sin servidor y se produce un error en el flujo de trabajo, es posible que no tenga permiso para crear un punto de conexión de VPC. Compruebe que el administrador le haya concedido permiso para crear un punto de conexión de VPC al permitir la acción `ec2:CreateVpcEndpoint`.

Para obtener una lista completa de los permisos necesarios para realizar todas las acciones de Amazon MSK, consulte AWS política gestionada: Amazon MSKFull Access.

No se puede actualizar KafkaVersionsList en la configuración de MSK

Al actualizar la KafkaVersionsListpropiedad del AWS::MSK::Configurationrecurso, se produce un error en la actualización y aparece el siguiente error.

```
Resource of type 'AWS::MSK::Configuration' with identifier '<identifierName>' already
 exists.
```

Al actualizar la `KafkaVersionsList` propiedad, AWS CloudFormation vuelve a crear una nueva configuración con la propiedad actualizada antes de eliminar la configuración anterior. La actualización de la AWS CloudFormation pila falla porque la nueva configuración usa el mismo nombre que la configuración existente. Esta actualización requiere la sustitución de un recurso. Para que la actualización se realice correctamente`KafkaVersionsList`, también debe actualizar la propiedad Nombre en la misma operación.

Además, si la configuración está asociada a algún clúster creado con AWS Management Console o AWS CLI, añada lo siguiente al recurso de configuración para evitar intentos fallidos de eliminación de recursos.

```
UpdateReplacePolicy: Retain
```

Cuando la actualización se realice correctamente, vaya a la consola de Amazon MSK y elimine la configuración anterior. Para obtener más información acerca de las configuraciones de MSK, consulte Configuración aprovisionada de Amazon MSK.

Mejores prácticas para corredores de Standard y Express

En esta sección se describen las mejores prácticas que deben seguir los corredores Standard y Express. Para obtener información sobre las prácticas recomendadas del Replicador Amazon MSK, consulte Prácticas recomendadas para utilizar el Replicador MSK.

Temas

- Prácticas recomendadas para agentes
- Mejores prácticas para los corredores de Express
- Prácticas recomendadas para los clientes de Apache Kafka

Prácticas recomendadas para agentes

En este tema se describen algunas de las prácticas recomendadas que debe seguir al utilizar Amazon MSK. Para obtener información sobre las prácticas recomendadas del Replicador Amazon MSK, consulte Prácticas recomendadas para utilizar el Replicador MSK.

Consideraciones del cliente

La disponibilidad y el rendimiento de su aplicación dependen no solo de la configuración del lado del servidor, sino también de la configuración del cliente.

- Configuración de sus clientes para alta disponibilidad. En un sistema distribuido como Apache Kafka, garantizar una alta disponibilidad es fundamental para mantener una infraestructura de mensajería fiable y tolerante a errores. Los agentes se desconectarán debido a eventos planificados o imprevistos, por ejemplo, actualizaciones, aplicaciones de parches, errores de hardware o problemas de red. Un clúster de Kafka tolera la presencia de un agente fuera de línea, por lo que los clientes de Kafka también deben administrar la conmutación por error de un agente sin problemas. Consulte todos los detalles en. Prácticas recomendadas para los clientes de Apache Kafka
- Asegúrese de que las cadenas de conexión del cliente incluyan al menos un agente de cada zona de disponibilidad. Tener varios agentes en la cadena de conexión de un cliente permite la conmutación por error cuando un agente específico está sin conexión para una actualización.

Para obtener información acerca de cómo obtener una cadena de conexión con varios corredores, consulte Obtención de agentes de arranque para un clúster de Amazon MSK.

- Realice pruebas de rendimiento para comprobar que las configuraciones de sus clientes le permiten cumplir sus objetivos de rendimiento.

Consideraciones del servidor

Ajuste el tamaño correcto del clúster: número de particiones por agente

La siguiente tabla muestra el número recomendado de particiones (incluidas las réplicas de seguidor y líder) por agente estándar. El número recomendado de particiones no se aplica y es una práctica recomendada en situaciones en las que se envía tráfico a todas las particiones temáticas aprovisionadas.

Tamaño del agente	Número recomendado de particiones (incluidas las réplicas de seguidor y líder) por agente	Número máximo de particion es que admiten operaciones de actualización
`kafka.t3.small`	300	300
`kafka.m5.large` o `kafka.m5.xlarge`	1 000	1500
`kafka.m5.2xlarge`	2000	3 000
`kafka.m5.4xlarge` `kafka.m5.8xlarge` `kafka.m5.12xlarge` , `kafka.m5.16xlarge` o `kafka.m5.24xlarge`	4000	6000
`kafka.m7g.large` o `kafka.m7g.xlarge`	1 000	1500
`kafka.m7g.2xlarge`	2000	3 000
`kafka.m7g.4xlarge` , `kafka.m7g.8xlarge` ,	4000	6000

Tamaño del agente	Número recomendado de particiones (incluidas las réplicas de seguidor y líder) por agente	Número máximo de particion es que admiten operaciones de actualización
`kafka.m7g.12xlarge` o `kafka.m7g.16xlarge`		

Si tiene un número elevado de particiones y un rendimiento reducido en los que tiene un mayor número de particiones, pero no envía tráfico a todas las particiones, puede empaquetar más particiones por agente, siempre y cuando haya realizado suficientes pruebas y pruebas de rendimiento para comprobar que el clúster se mantiene en buen estado con un mayor número de particiones. Si el número de particiones por agente supera el valor máximo permitido y el clúster se sobrecarga, se le impedirá realizar las siguientes operaciones:

- Actualizar la configuración de un clúster
- Actualización del clúster a un tamaño más pequeño del agente
- Asocie un AWS Secrets Manager secreto de a un clúster que tenga autenticación SASL/SCRAM

Un número elevado de particiones también puede hacer que falten métricas de Kafka en el raspado de CloudWatch Prometheus.

Para obtener instrucciones acerca de cómo elegir el número de particiones, consulte Apache Kafka Supports 200K Partitions Per Cluster. Sin embargo, también se recomienda que realice sus propias pruebas para determinar el tamaño de instancia adecuado para sus agentes. Para obtener más información sobre los distintos tamaños de agente, consulte the section called “Tipos de agente”.

Ajuste el tamaño correcto de su clúster: número de agentes estándar por clúster

Para determinar el número adecuado de agentes de su clúster aprovisionado de MSK y comprender los costos, consulte la hoja de cálculo MSK Sizing and Pricing. Esta hoja de cálculo proporciona una estimación del tamaño de un clúster aprovisionado de MSK y los costos asociados de Amazon MSK en comparación con un clúster Apache Kafka similar, administrado por cuenta propia, basado en Apache Kafka. EC2 Para obtener más información acerca de los parámetros de entrada en la hoja de cálculo, pasa el puntero del ratón por encima de las descripciones de los parámetros. Las estimaciones proporcionadas en esta hoja son moderadas y proporcionan un punto de partida para

un nuevo clúster aprovisionado de MSK. El rendimiento, el tamaño y los costos del clúster dependen del caso de uso, por lo que le recomendamos que los compruebe realizando pruebas reales.

Para entender cómo la infraestructura subyacente afecta al rendimiento de Apache Kafka, consulte Best practices for right-sizing your Apache Kafka clusters to optimize performance and cost en el AWS Blog de macrodatos de. La entrada del blog proporciona información sobre cómo dimensionar los clústeres para cumplir con los requisitos de rendimiento, disponibilidad y latencia. También proporciona respuestas a preguntas como cuándo se debe escalar o desescalar verticalmente, así como también orientación sobre cómo verificar continuamente el tamaño de los clústeres de producción. Para obtener información sobre los clústeres basados en almacenamiento en niveles, consulte Prácticas recomendadas para ejecutar cargas de trabajo de producción con el almacenamiento en niveles de Amazon MSK.

Optimización del rendimiento de los clústeres para instancias m5.4xl, m7g.4xl o más grandes

Cuando utilice instancias m5.4xl, m7g.4xl o más grandes, puede ajustar las configuraciones num.io.threads y num.network.threads para optimizar el rendimiento del MSK.

Num.io.threads es el número de subprocesos que utiliza un agente estándar para procesar las solicitudes. Si se agregan más subprocesos, hasta el número de núcleos de CPU compatibles con el tamaño de la instancia, se mejora el rendimiento del clúster.

Num.network.threads es el número de subprocesos que el agente estándar utiliza para recibir todas las solicitudes entrantes y devolver las respuestas. Los subprocesos de red colocan las solicitudes entrantes en una cola de solicitudes para que io.threads las procese. Si se configura num.network.threads en la mitad del número de núcleos de CPU compatibles con el tamaño de instancia, se podrá utilizar al máximo el nuevo tamaño de instancia.

> **Important**
>
> No aumente num.network.threads sin aumentar primero num.io.threads, ya que esto puede provocar una congestión relacionada con la saturación de las colas.

Configuración recomendada

Tamaño de instancia	Valor recomendado para num.io.threads	Valor recomendado para num.network.threads
m5.4xl	16	8

Tamaño de instancia	Valor recomendado para num.io.threads	Valor recomendado para num.network.threads
m5.8xl	32	16
m5.12xl	48	24
m5.16xl	64	32
m5.24xl	96	48
m7g.4xlarge	16	8
m7g.8xlarge	32	16
m7g.12xlarge	48	24
m7g.16xlarge	64	32

Utilice la última versión de Kafka AdminClient para evitar problemas de discordancia en los ID de los temas

El ID de un tema se pierde (Error: does not match the topic Id for partition) cuando se utiliza una AdminClient versión de Kafka anterior `--zookeeper` a la 2.8.0 con el indicador para aumentar o reasignar las particiones de temas de un clúster aprovisionado de MSK que utilice la versión 2.8.0 o superior de Kafka. Tenga en cuenta que el indicador `--zookeeper` está obsoleto en la versión 2.5 de Kafka y se elimina a partir de la versión 3.0 de Kafka. Consulte Upgrading to 2.5.0 from any version 0.8.x through 2.4.x.

Para evitar problemas de discordancia en los ID de los temas, utilice un cliente de la versión 2.8.0 o superior de Kafka para las operaciones de administración de Kafka. Como alternativa, los clientes de la versión 2.5 y posterior pueden usar el indicador `--bootstrap-servers` en lugar del indicador `--zookeeper`.

Crear clústeres de alta disponibilidad

Siga las recomendaciones a continuación para que sus clústeres aprovisionados de MSK tengan una alta disponibilidad durante una actualización (por ejemplo, al actualizar el tamaño del agente o la versión de Apache Kafka) o cuando Amazon MSK sustituya a un agente.

- Configure un clúster con tres zonas de disponibilidad.
- Asegúrese de que el factor de replicación (RF) sea de al menos 3. Tenga en cuenta que un RF de 1 puede provocar que las particiones estén sin conexión durante una actualización sucesiva, y un RF de 2 puede provocar la pérdida de datos.
- Establezca el mínimo de réplicas en sincronización (MiniSR) como máximo RF - 1. Un miniSR igual a RF puede impedir que se produzca en el clúster durante una actualización sucesiva. Un miniSR de 2 permite que los temas replicados de tres vías estén disponibles cuando una réplica está fuera de línea.

Supervisisión del uso de CPU

Amazon MSK recomienda encarecidamente que mantenga el uso de la CPU por parte de sus agentes (definido como`CPU User + CPU System`) por debajo del 60%. Esto garantiza que su clúster conserve suficiente espacio de CPU para gestionar los eventos operativos, como los fallos de los intermediarios, la aplicación de parches y las actualizaciones sucesivas.

Apache Kafka puede redistribuir la carga de la CPU entre los agentes del clúster cuando sea necesario. Por ejemplo, cuando Amazon MSK detecta un error de un agente y lo resuelve, realiza un mantenimiento automático, como la aplicación de parches. Del mismo modo, cuando un usuario solicita un cambio de tamaño del agente o una actualización de versión, Amazon MSK inicia flujos de trabajo continuos que desconectan los agentes uno por uno. Cuando los agentes con particiones principales se desconectan, Apache Kafka reasigna el liderazgo de la partición para redistribuir el trabajo entre los demás agentes del clúster. Si sigue esta práctica recomendada, se asegura de que la CPU tenga suficiente espacio para tolerar estos eventos operativos.

> **Note**
>
> Al supervisar el uso de la CPU, tenga en cuenta que el uso total de la CPU incluye más de `CPU User` y`CPU System`. Otras categorías, como,`iowait`, `yirq`, `softirqsteal`, también contribuyen a la actividad general de la CPU. Por lo tanto, la CPU inactiva no siempre es igual a`100% - CPU User - CPU System`.

Puedes usar las matemáticas CloudWatch métricas de Amazon para crear una métrica compuesta (`CPU User + CPU System`) y configurar una alarma para que se active cuando el uso medio supere el 60%. Cuando se active, considere escalar el clúster mediante una de las siguientes opciones:

- Opción 1 (recomendada): actualice el tamaño de su agente al siguiente tamaño más grande. Por ejemplo, si el tamaño actual es `kafka.m5.large`, actualice el clúster para que utilice `kafka.m5.xlarge`. Tenga en cuenta que, al actualizar el tamaño del agente del clúster, Amazon MSK desconecta a los agentes de forma continua y reasigna temporalmente el liderazgo de la partición a otros agentes. Por lo general, una actualización de tamaño tarda entre 10 y 15 minutos por agente.
- Opción 2: si hay temas en los que todos los mensajes provienen de productores que realizan escrituras siguiendo un sistema rotativo (en otras palabras, los mensajes no están codificados y el orden no es importante para los consumidores), amplíe su clúster agregando agentes. Agregue también particiones a los temas existentes con el mayor rendimiento. A continuación, use `kafka-topics.sh --describe` para asegurarse de que las particiones recién agregadas se asignen a los nuevos agentes. La principal ventaja de esta opción en comparación con la anterior es que permite administrar los recursos y los costos de forma más detallada. Además, puede utilizar esta opción si la carga de la CPU supera con creces el 60 %, ya que esta forma de escalado no suele provocar un aumento de la carga para los agentes existentes.
- Opción 3: amplíe su clúster aprovisionado de MSK agregando agentes y, a continuación, reasigne las particiones existentes mediante la herramienta de reasignación de particiones denominada. `kafka-reassign-partitions.sh` Sin embargo, si usa esta opción, el clúster necesitará gastar recursos para replicar los datos de un agente a otro después de reasignar las particiones. En comparación con las dos opciones anteriores, esto puede aumentar significativamente la carga del clúster al principio. En consecuencia, Amazon MSK no recomienda usar esta opción cuando el uso de la CPU sea superior al 70 %, ya que la replicación provoca una carga de CPU y tráfico de red adicionales. Amazon MSK solo recomienda usar esta opción si las dos opciones anteriores no son factibles.

Otras recomendaciones:

- Supervise el uso total de la CPU por agente como proxy para la distribución de la carga. Si los agentes utilizan la CPU de manera uniforme, podría ser una señal de que la carga no está distribuida de manera uniforme dentro del clúster. Recomendamos utilizar Cruise Control para administrar de forma continua la distribución de la carga mediante la asignación de particiones.
- Supervise la latencia de producción y consumo. La latencia de producción y consumo puede aumentar linealmente con el uso de la CPU.
- Intervalo de raspado de JMX: si habilita la supervisión abierta con la característica Prometheus, se recomienda utilizar un intervalo de raspado de 60 segundos o más (scrape_interval: 60s) para la

configuración de su host de Prometheus (prometheus.yml). Reducir el intervalo de raspado puede provocar un uso elevado de la CPU en el clúster.

Monitorear el espacio en disco

Para evitar quedarse sin espacio en disco para los mensajes, cree una CloudWatch alarma que vigile la `KafkaDataLogsDiskUsed` métrica. Cuando el valor de esta métrica alcance o supere el 85 %, realice una o varias de las siguientes acciones:

- Utilice the section called “Escalado automático para clústeres”. También puede aumentar manualmente el almacenamiento del agente, tal y como se describe en the section called “Escalado manual”.
- Reduzca el periodo de retención de mensajes o el tamaño del registro. Para obtener información sobre cómo hacerlo, consulte the section called “Ajuste los parámetros de retención de datos”.
- Elimine los temas que no se utilicen.

Para obtener información sobre cómo configurar y usar las alarmas, consulta Uso de Amazon CloudWatch Alarms. Para obtener una lista completa de las métricas de Amazon MSK, consulte the section called “Supervisión de un clúster”.

Ajuste los parámetros de retención de datos

El consumo de los mensajes no los elimina del registro. Para liberar espacio en disco de manera regular, puede especificar explícitamente un periodo de retención, que es el tiempo que permanecen los mensajes en el registro. También puede especificar el tamaño del registro de una retención. Cuando se alcanza el periodo de retención o el tamaño del registro de retención, Apache Kafka comienza a eliminar los segmentos inactivos del registro.

Para especificar una política de retención en el nivel del clúster, establezca uno o varios de los siguientes parámetros: `log.retention.hours`, `log.retention.minutes`, `log.retention.ms` o `log.retention.bytes`. Para obtener más información, consulte the section called “Configuraciones personalizadas de Amazon MSK”.

También puede especificar los parámetros de retención en el nivel de tema:

- Para especificar un periodo de retención por tema, utilice el siguiente comando.

```
kafka-configs.sh --bootstrap-server $bs --alter --entity-type topics --entity-
name TopicName --add-config retention.ms=DesiredRetentionTimePeriod
```

- Para especificar un tamaño de registro de retención por tema, utilice el siguiente comando.

```
kafka-configs.sh --bootstrap-server $bs --alter --entity-type topics --entity-
name TopicName --add-config retention.bytes=DesiredRetentionLogSize
```

Los parámetros de retención que especifique en el nivel del tema tienen preferencia sobre los parámetros del nivel del clúster.

Aceleración de la recuperación de registros después de un cierre incorrecto

Tras un cierre incorrecto, un agente puede tardar un poco en reiniciarse, al igual que hace con la recuperación de registros. De forma predeterminada, Kafka solo usa un subproceso por directorio de registro para realizar esta recuperación. Por ejemplo, si tiene miles de particiones, la recuperación del registro puede tardar horas en completarse. Para acelerar la recuperación de registros, se recomienda aumentar el número de subprocesos mediante la propiedad de configuración `num.recovery.threads.per.data.dir`. Puede configurarla en el número de núcleos de CPU.

Supervisión de la memoria de Apache Kafka

Se recomienda supervisar la memoria que utiliza Apache Kafka. De lo contrario, es posible que el clúster deje de estar disponible.

Para determinar la cantidad de memoria que utiliza Apache Kafka, puede supervisar la métrica `HeapMemoryAfterGC`. `HeapMemoryAfterGC` es el porcentaje de la memoria dinámica total que se utiliza después de la recopilación de elementos no utilizados. Le recomendamos que cree una CloudWatch alarma que actúe cuando `HeapMemoryAfterGC` aumente por encima del 60%.

Los pasos que puede realizar para reducir el uso de memoria varían. Dependen de la forma en que se configure Apache Kafka. Por ejemplo, si utiliza la entrega transaccional de mensajes, puede reducir el valor de `transactional.id.expiration.ms` de la configuración de Apache Kafka de `604800000` ms a `86400000` ms (de 7 días a 1 día). Esto reduce la huella de memoria de cada transacción.

No agregue a agentes que no sean de MSK

En el ZooKeeper caso de los clústeres aprovisionados de MSK, si utiliza ZooKeeper comandos de Apache para agregar agentes, estos agentes no se agregan a su clúster aprovisionado de MSK y su Apache ZooKeeper contendrá la información incorrecta sobre el clúster. Esto podría provocar la pérdida de datos. Para obtener información sobre las operaciones de clúster aprovisionadas por MSK compatibles, consulte. the section called “Características y conceptos clave”

Habilitar el cifrado en tránsito

Para obtener información sobre el cifrado en tránsito y cómo habilitarlo, consulte the section called “Cifrado en tránsito de Amazon MSK”.

Reasignar particiones

Para mover particiones a distintos agentes en el mismo clúster aprovisionado de MSK, puede usar la herramienta de reasignación de particiones denominada. `kafka-reassign-partitions.sh` Le recomendamos que no reasigne más de 10 particiones en una sola `kafka-reassign-partitions` llamada para garantizar la seguridad de las operaciones. Por ejemplo, después de agregar nuevos agentes para expandir un clúster o para mover las particiones y eliminar agentes, puede reequilibrar el clúster reasignando las particiones a los nuevos agentes. Para obtener información acerca de cómo agregar agentes a un clúster aprovisionado de MSK, consulte. the section called “Amplíe un clúster” Para obtener información acerca de cómo eliminar agentes de un clúster aprovisionado de MSK, consulte. the section called “Eliminación de un agente” Para obtener información acerca de la herramienta de reasignación de particiones, consulte Expansión de su clúster en la documentación de Apache Kafka.

Mejores prácticas para los corredores de Express

En este tema se describen algunas de las mejores prácticas que se deben seguir al utilizar corredores de Express. Los corredores Express vienen preconfigurados para ofrecer una alta disponibilidad y durabilidad. De forma predeterminada, sus datos se distribuyen en tres zonas de disponibilidad, la replicación siempre se establece en 3 y la réplica sincronizada mínima siempre se establece en 2. Sin embargo, aún hay algunos factores que se deben tener en cuenta para optimizar la confiabilidad y el rendimiento del clúster.

Consideraciones del lado del cliente

La disponibilidad y el rendimiento de su aplicación dependen no solo de la configuración del lado del servidor, sino también de la configuración del cliente.

- Configure sus clientes para una alta disponibilidad. En un sistema distribuido como Apache Kafka, garantizar una alta disponibilidad es fundamental para mantener una infraestructura de mensajería fiable y tolerante a errores. Los corredores se desconectarán por eventos planificados y no planificados, por ejemplo, por actualizaciones, parches, fallas de hardware o problemas de red. Un clúster de Kafka tolera la presencia de un agente fuera de línea, por lo que los clientes de Kafka también deben administrar la conmutación por error de un agente sin problemas. Consulte todos los detalles en las recomendaciones de mejores prácticas para los clientes de Apache Kafka.
- Realice pruebas de rendimiento para comprobar que las configuraciones de sus clientes le permiten cumplir sus objetivos de rendimiento incluso cuando reiniciamos los corredores en momentos de máxima carga. Puede reiniciar los corredores de su clúster desde la consola MSK o mediante APIs MSK.

Consideraciones del lado del servidor

Ajuste el tamaño correcto de su clúster: número de agentes por clúster

Elegir el número de corredores para su clúster basado en Express es fácil. Cada bróker Express viene con una capacidad de rendimiento definida para la entrada y la salida. Debe utilizar esta capacidad de rendimiento como medio principal para dimensionar el clúster (y, a continuación, tener en cuenta otros factores, como el número de particiones y conexiones, que se analizan a continuación).

Por ejemplo, si su aplicación de streaming necesita un 45% MBps de capacidad de entrada (escritura) de datos y un 90% de salida (lectura) de MBps datos, simplemente puede utilizar 3 corredores express.m7g.large para satisfacer sus necesidades de rendimiento. Cada bróker de express.m7g.large gestionará el 15% de las entradas y el 30% de las salidas. MBps MBps Consulte en la siguiente tabla nuestros límites de rendimiento recomendados para cada tamaño de bróker Express. Si su rendimiento supera los límites recomendados, es posible que experimente una disminución del rendimiento y que deba reducir el tráfico o escalar el clúster. Si su rendimiento supera los límites recomendados y alcanza la cuota por intermediario, MSK reducirá el tráfico de sus clientes para evitar una mayor sobrecarga.

También puede utilizar nuestra hoja de cálculo de tamaño y precios de MSK para evaluar varios escenarios y tener en cuenta otros factores, como el número de particiones.

Rendimiento máximo recomendado por bróker

Tamaño de instancia	Entrada () MBps	Salida () MBps
express.m7g.large	15.6	31,2
express.m7g.xlarge	31,2	62,5
express.m7g.2xlarge	62,5	125,0
express.m7g.4xlarge	124,9	249,8
express.m7g.8xlarge	250,0	500,0
express.m7g.12xlarge	375,0	750,0
express.m7g.16xlarge	500,0	1000,0

Supervisisión del uso de CPU

Le recomendamos que mantenga el uso total de la CPU de sus corredores (definido como usuario de CPU más sistema de CPU) por debajo del 60%. Cuando tenga disponible al menos el 40 % de la CPU total del clúster, Apache Kafka podrá redistribuir la carga de la CPU entre los agentes del clúster cuando sea necesario. Esto puede ser necesario debido a eventos planificados o imprevistos. Un ejemplo de evento planificado es una actualización de la versión de un clúster durante la cual MSK actualiza los corredores de un clúster reiniciándolos uno por uno. Un ejemplo de evento no planificado es un fallo de hardware en un bróker o, en el peor de los casos, un fallo en una zona de disponibilidad que afecte a todos los agentes de una zona de disponibilidad. Cuando los corredores con réplicas de leads de partición se desconectan, Apache Kafka reasigna el liderazgo de la partición para redistribuir el trabajo entre los demás corredores del clúster. Si sigue esta práctica recomendada, puede asegurarse de tener suficiente espacio de CPU en su clúster para tolerar eventos operativos como estos.

Puedes usar el uso de expresiones matemáticas con CloudWatch métricas en la Guía del CloudWatch usuario de Amazon para crear una métrica compuesta que sea Usuario de CPU y Sistema de CPU. Configure una alarma que se active cuando la métrica compuesta alcance una utilización media de la CPU del 60 %. Cuando se active esta alarma, escale el clúster mediante una de las siguientes opciones:

- Opción 1: actualice el tamaño de su corredor al tamaño siguiente más grande. Tenga en cuenta que, al actualizar el tamaño del agente del clúster, Amazon MSK desconecta a los agentes de forma continua y reasigna temporalmente el liderazgo de la partición a otros agentes.
- Opción 2: Amplíe su clúster añadiendo intermediarios y, a continuación, reasignando las particiones existentes mediante la herramienta de reasignación de particiones denominada. `kafka-reassign-partitions.sh`

Otras recomendaciones

- Supervise el uso total de la CPU por agente como proxy para la distribución de la carga. Si los intermediarios utilizan la CPU de manera uniforme, podría ser una señal de que la carga no está distribuida uniformemente dentro del clúster. Recomendamos utilizar Cruise Control para gestionar de forma continua la distribución de la carga mediante la asignación de particiones.
- Supervise la latencia de producción y consumo. La latencia de producción y consumo puede aumentar linealmente con el uso de la CPU.
- Intervalo de raspado de JMX: si habilita la supervisión abierta con la función Prometheus, se recomienda utilizar un intervalo de raspado de 60 segundos o más () `scrape_interval: 60s` para la configuración de su host de Prometheus (). `prometheus.yml` Reducir el intervalo de raspado puede provocar un uso elevado de la CPU en el clúster.

Ajuste el tamaño correcto de su clúster: número de particiones por broker Express

La siguiente tabla muestra la cantidad recomendada de particiones (incluidas las réplicas líder y seguidora) por agente de Express. El número recomendado de particiones no se aplica y es una práctica recomendada en situaciones en las que se envía tráfico a todas las particiones temáticas aprovisionadas.

Tamaño del agente	Número recomendado de particiones (incluidas las réplicas de seguidor y líder) por agente	Número máximo de particion es que admiten operaciones de actualización
`express.m7g.large`	1 000	1500
`express.m7g.xlarge`		

Tamaño del agente	Número recomendado de particiones (incluidas las réplicas de seguidor y líder) por agente	Número máximo de particion es que admiten operaciones de actualización
express.m7g.2xlarge	2000	3 000
express.m7g.4xlarge	4000	6000
express.m7g.8xlarge		
express.m7g.12xlarge		
express.m7g.16xlarge		

Si tiene un número elevado de particiones y un rendimiento reducido en los que tiene un mayor número de particiones, pero no envía tráfico a todas las particiones, puede empaquetar más particiones por agente, siempre y cuando haya realizado suficientes pruebas y pruebas de rendimiento para comprobar que el clúster se mantiene en buen estado con un mayor número de particiones. Si el número de particiones por agente supera el valor máximo permitido y el clúster se sobrecarga, no podrá realizar las siguientes operaciones:

- Actualizar la configuración de un clúster
- Actualización del clúster a un tamaño más pequeño del agente
- Asocie un AWS Secrets Manager secreto a un clúster que tenga autenticación SASL/SCRAM

Un clúster sobrecargado con un gran número de particiones también puede hacer que falten métricas de Kafka en el CloudWatch raspado de Prometheus.

Para obtener instrucciones acerca de cómo elegir el número de particiones, consulte Apache Kafka Supports 200K Partitions Per Cluster. Sin embargo, también se recomienda que realice sus propias pruebas para determinar el tamaño de instancia adecuado para sus agentes. Para obtener más información sobre los distintos tamaños de agente, consulte Tamaños del agente de Amazon MSK.

Supervise el recuento de conexiones

Las conexiones de los clientes con sus corredores consumen recursos del sistema, como la memoria y la CPU. En función de su mecanismo de autenticación, debe realizar un seguimiento

para asegurarse de que se encuentra dentro de los límites aplicables. Para gestionar los reintentos en caso de conexiones fallidas, puede configurar el parámetro de configuración `reconnect.backoff.ms` en el lado del cliente. Por ejemplo, si desea que un cliente vuelva a intentar conectarse después de 1 segundo, `reconnect.backoff.ms` configúrelo en. `1000` Para obtener más información sobre la configuración de los reintentos, consulte la documentación de Apache Kafka.

Dimensión	Cuota
Número máximo de conexiones TCP por intermediario (control de acceso de IAM)	3 000
Número máximo de conexiones TCP por intermediario (IAM)	100 por segundo
Número máximo de conexiones TCP por intermediario (no IAM)	MSK no impone límites de conexión para la autenticación que no sea de IAM. Sin embargo, debe supervisar otras métricas, como el uso de la CPU y la memoria, para asegurarse de no sobrecargar el clúster debido a un exceso de conexiones.

Reasignar particiones

Para mover particiones a distintos agentes del mismo clúster aprovisionado por MSK, puedes usar la herramienta de reasignación de particiones denominada. `kafka-reassign-partitions.sh` Le recomendamos que no reasigne más de 20 particiones en una sola `kafka-reassign-partitions` llamada para garantizar la seguridad de las operaciones. Por ejemplo, después de agregar nuevos agentes para expandir un clúster o para mover las particiones y eliminar agentes, puede reequilibrar el clúster reasignando las particiones a los nuevos agentes. Para obtener información sobre cómo añadir agentes a un clúster aprovisionado de MSK, consulte. the section called “Amplíe un clúster” Para obtener información sobre cómo eliminar intermediarios de un clúster aprovisionado de MSK, consulte. the section called “Eliminación de un agente” Para obtener información acerca de la herramienta de reasignación de particiones, consulte Expansión de su clúster en la documentación de Apache Kafka.

Prácticas recomendadas para los clientes de Apache Kafka

Al trabajar con Apache Kafka y Amazon MSK, es importante configurar correctamente tanto el cliente como el servidor para lograr un rendimiento y una fiabilidad óptimos. En esta guía, se ofrecen recomendaciones sobre las prácticas recomendadas de configuración del cliente para Amazon MSK.

Para obtener información sobre las prácticas recomendadas del Replicador Amazon MSK, consulte Prácticas recomendadas para utilizar el Replicador MSK. Para conocer las mejores prácticas de los brókers Standard y Express, consulteMejores prácticas para corredores de Standard y Express.

Temas

- Disponibilidad del cliente de Apache Kafka
- rendimiento del cliente de Apache Kafka
- Supervisión de clientes de Kafka

Disponibilidad del cliente de Apache Kafka

En un sistema distribuido como Apache Kafka, garantizar una alta disponibilidad es fundamental para mantener una infraestructura de mensajería fiable y tolerante a errores. Los agentes se desconectarán debido a eventos planificados o imprevistos, como actualizaciones, aplicaciones de parches, errores de hardware o problemas de red. Un clúster de Kafka tolera la presencia de un agente fuera de línea, por lo que los clientes de Kafka también deben administrar la conmutación por error de un agente sin problemas. Para garantizar la alta disponibilidad de los clientes de Kafka, recomendamos las siguientes prácticas recomendadas.

Disponibilidad del productor

- Establezca `retries` para indicar al productor que intente enviar nuevamente los mensajes con errores durante la conmutación por error del agente. Recomendamos que el valor utilizado sea un número entero máximo o un valor alto similar para la mayoría de los casos de uso. De lo contrario, se interrumpirá la alta disponibilidad de Kafka.
- Establezca `delivery.timeout.ms` para especificar el límite superior del tiempo total entre el envío de un mensaje y la recepción de una confirmación por parte del agente. Esto debe reflejar los requisitos empresariales respecto al tiempo de validez de un mensaje. Establezca un límite de tiempo lo suficientemente alto como para permitir que se realice la cantidad necesaria de reintentos para completar la conmutación por error. Recomendamos que se utilice un valor de 60 segundos o superior para la mayoría de los casos de uso.

- Establezca `request.timeout.ms` al tiempo máximo que una solicitud debe esperar antes de que se intente el reenvío. Recomendamos que se utilice un valor de 10 segundos o superior para la mayoría de los casos de uso.
- Establezca `retry.backoff.ms` para configurar el retraso entre los reintentos para evitar una avalancha de reintentos y un impacto en la disponibilidad. Recomendamos que se utilice un valor mínimo de 200 ms para la mayoría de los casos de uso.
- Establezca `acks=all` para configurar una alta durabilidad; esto debe alinearse con una configuración del servidor de `RF=3` y `min.isr=2` para garantizar que todas las particiones del ISR reconozcan la escritura. Cuando un solo agente está fuera de línea, este es el `min.isr`, es decir, 2.

Disponibilidad de los consumidores

- Establezca `auto.offset.reset` en `latest` inicialmente para los grupos de consumidores nuevos o creados nuevamente. Esto evita el riesgo de aumentar la carga de clústeres al consumir todo el tema.
- Establezca `auto.commit.interval.ms` cuando utilice `enable.auto.commit`. Recomendamos que se utilice un valor mínimo de 5 segundos para la mayoría de los casos de uso, para evitar el riesgo de una carga adicional.
- Implemente la gestión de excepciones en el código de procesamiento de mensajes del consumidor para gestionar los errores transitorios, por ejemplo, un disyuntor o una suspensión con un retroceso exponencial. Si no lo hace, puede producirse un error en la aplicación, lo que puede provocar un reequilibrio excesivo.
- Establezca `isolation.level` para controlar cómo se leen los mensajes transaccionales:

  Recomendamos que siempre se establezca `read_uncommitted` de forma implícita y predeterminada. Esto no aparece en algunas implementaciones de clientes.

  Recomendamos que utilice un valor de `read_uncommitted` al utilizar un almacenamiento en niveles.
- Establezca `client.rack` para usar la lectura de réplica más cercana. Recomendamos que se establezca en `az id` para minimizar los costos del tráfico de red y la latencia. Consulte Reduce network traffic costs of your Amazon MSK consumers with rack awareness.

Reequilibrios de los consumidores

- Establezca `session.timeout.ms` en un valor superior al tiempo de inicio de una aplicación, incluida cualquier fluctuación de inicio implementada. Recomendamos que se utilice un valor de 60 segundos para la mayoría de los casos de uso.
- Establezca `heartbeat.interval.ms` para afinar la forma en que el coordinador del grupo considera que un consumidor está en buen estado. Recomendamos que se utilice un valor de 10 segundos para la mayoría de los casos de uso.
- Establezca un enlace de cierre en su aplicación para cerrar sin problemas al consumidor en SIGTERM, en lugar de depender de los tiempos de espera de la sesión para identificar cuándo un consumidor abandona un grupo. Las aplicaciones de Kstream pueden establecer `internal.leave.group.on.close` en un valor de `true`.
- Establezca `group.instance.id` en un valor distinto dentro del grupo de consumidores. Idealmente, un nombre de host, task-id o pod-id. Recomendamos que siempre establezca esto para lograr comportamientos más deterministas y una mejor correlación de registros entre el cliente y el servidor durante la resolución de problemas.
- Establezca `group.initial.rebalance.delay.ms` en un valor que se alinee con el tiempo medio de implementación. Esto detiene los reequilibrios continuos durante la implementación.
- Establezca `partition.assignment.strategy` para utilizar asignadores fijos. Recomendamos que sea `StickyAssignor` o `CooperativeStickyAssignor`.

rendimiento del cliente de Apache Kafka

Para garantizar un alto rendimiento de los clientes de Kafka, recomendamos que se sigan estas prácticas recomendadas.

rendimiento del productor

- Establezca `linger.ms` para controlar el tiempo que un productor espera a que se llene un lote. Los lotes más pequeños son costosos desde el punto de vista computacional para Kafka, ya que generan más subprocesos y operaciones de E/S simultáneamente. Recomendamos que se utilicen los siguientes valores.

  Un valor mínimo de 5 ms para todos los casos de uso, incluida la baja latencia.

  Recomendamos que se utilice un valor más alto de 25 ms para la mayoría de los casos de uso.

Recomendamos que nunca se utilice un valor de cero en los casos de uso con una latencia baja. (Por lo general, un valor de cero provoca latencia sin importar la sobrecarga de E/S).

- Establezca `batch.size` para controlar el tamaño del lote que se envía al clúster. Recomendamos que lo aumente a un valor de 64 KB o 128 KB.
- Establezca `buffer.memory` cuando utilice lotes de mayor tamaño. Recomendamos que se utilice un valor de 64 MB para la mayoría de los casos de uso.
- Establezca `send.buffer.bytes` para controlar el búfer TCP que se utiliza para recibir bytes. Recomendamos que se utilice un valor de -1 para que el sistema operativo pueda administrar este búfer cuando ejecute un productor en una red de alta latencia.
- Establezca compression.type para controlar la compresión de los lotes. Recomendamos que lz4 o zstd ejecuten un productor en una red de alta latencia.

rendimiento de los consumidores

- Establezca `fetch.min.bytes` para controlar el tamaño mínimo de recuperación que sea válido para reducir la cantidad de recuperaciones y la carga del clúster.

  Recomendamos que se utilice un valor mínimo de 32 bytes para todos los casos de uso.

  Recomendamos que se utilice un valor más alto de 128 bytes para la mayoría de los casos de uso.
- Establezca fetch.max.wait.ms para determinar cuánto tiempo esperará el consumidor antes de que se ignore fetch.min.bytes. Recomendamos que se utilice un valor de 1000 ms para la mayoría de los casos de uso.
- Recomendamos que el número de consumidores sea, al menos, igual al número de particiones para mejorar el paralelismo y la resiliencia. En algunas situaciones es posible que decida tener menos consumidores que el número de particiones para temas de bajo rendimiento.
- Establezca `receive.buffer.bytes` para controlar el búfer TCP que se utiliza para recibir bytes. Recomendamos que se utilice un valor de -1 para que el sistema operativo pueda administrar este búfer cuando ejecute un consumidor en una red de alta latencia.

Conexiones de clientes

El ciclo de vida de las conexiones tiene un costo computacional y de memoria en un clúster Kafka. Si se crean demasiadas conexiones a la vez, se produce una carga que puede afectar a la disponibilidad de un clúster de Kafka. Este impacto en la disponibilidad suele provocar que las

aplicaciones creen aún más conexiones, lo que provoca un error en cascada y una interrupción total. Se puede lograr un número elevado de conexiones si se crean a un ritmo razonable.

Recomendamos que se utilicen las siguientes medidas de mitigación para administrar las altas tasas de creación de conexiones:

- Asegúrese de que el mecanismo de implementación de sus aplicaciones no reinicie a todos los productores o consumidores a la vez, sino preferiblemente en lotes más pequeños.
- En el nivel de aplicación, el desarrollador debe garantizar que se produzca una fluctuación aleatoria (suspensión aleatoria) antes de crear un cliente de administración, un cliente productor o un cliente consumidor.
- En SIGTERM, al cerrar la conexión, se debe ejecutar una suspensión aleatoria para garantizar que los clientes de Kafka no se cierren todos al mismo tiempo. La suspensión aleatoria debe estar dentro del tiempo de espera antes de que se produzca SIGKILL.

  Example Ejemplo A (Java)

  ```
  sleepInSeconds(randomNumberBetweenOneAndX);
                          this.kafkaProducer = new KafkaProducer<>(this.props);
  ```

  Example Ejemplo B (Java)

  ```
  Runtime.getRuntime().addShutdownHook(new Thread(() -> {
      sleepInSeconds(randomNumberBetweenOneAndTwentyFive);
      kafkaProducer.close(Duration.ofSeconds(5));
  });
  ```

- En el nivel de aplicación, el desarrollador debe garantizar que los clientes se creen solo una vez por aplicación y con un patrón único. Por ejemplo, cuando se usa Lambda, el cliente debe crearse en un ámbito global y no en el controlador de métodos.
- Recomendamos que se controle la cantidad de conexiones para que se mantenga estable. creation/close/shiftLa conexión son habituales durante las implementaciones y la conmutación por error de los agentes.

Supervisión de clientes de Kafka

Supervisar a los clientes de Kafka es fundamental para mantener el buen estado y la eficiencia de su ecosistema de Kafka. Sin importar si es un administrador de Kafka, un desarrollador o un miembro

del equipo de operaciones, la habilitación de las métricas del cliente es crucial para comprender el impacto empresarial durante eventos planificados y no planificados.

Recomendamos que se utilice el mecanismo de captura de métricas de preferencia para supervisar las siguientes métricas del cliente.

Cuando genere tickets de soporte con AWS, incluya cualquier valor anormal que observe durante el incidente. Incluya también una muestra de los registros de la aplicación del cliente en los que se detallen los errores (no las advertencias).

Métricas de productores

- byte-rate
- record-send-rate
- records-per-request-avg
- acks-latency-avg
- request-latency-avg
- request-latency-max
- record-error-rate
- record-retry-rate
- error-rate

> **Note**
>
> No se preocupe por los errores transitorios en los reintentos, ya que forman parte del protocolo de Kafka para gestionar problemas transitorios, como la conmutación por error de las particiones principales o las retransmisiones de la red. `record-send-rate`confirmará si los productores siguen realizando nuevos intentos.

Métricas de consumidores

- records-consumed-rate
- bytes-consumed-rate
- fetch-rate
- records-lag-max

- record-error-rate
- fetch-error-rate
- poll-rate
- rebalance-latency-avg
- commit-rate

> **Note**
>
> Las altas tasas de recuperación y de confirmación provocarán una carga innecesaria en el clúster. Lo ideal es realizar las solicitudes en lotes más grandes.

Métricas comunes

- connection-close-rate
- connection-creation-rate
- connection-count

> **Note**
>
> La creación o finalización de alta conexión provocará una carga innecesaria en el clúster.

¿Qué es MSK sin servidor?

> **Note**
>
> MSK sin servidor está disponible en las regiones de Este de EE. UU. (Ohio), Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón), Canadá (centro), Asia-Pacífico (Bombay), Asia-Pacífico (Singapur), Asia-Pacífico (Sídney), Asia-Pacífico (Tokio), Asia-Pacífico (Seúl), Europa (Fráncfort), Europa (Estocolmo), Europa (Irlanda), Europa (París) y Europa (Londres).

MSK sin servidor es un tipo de clúster para Amazon MSK que le permite ejecutar Apache Kafka sin tener que administrar ni escalar la capacidad del clúster. Aprovisiona y escala automáticamente la capacidad y, al mismo tiempo, administra las particiones de su tema, de modo que pueda transmitir datos sin tener que pensar en dimensionar o escalar los clústeres correctamente. MSK sin servidor ofrece un modelo de precios basado en el rendimiento, por lo que solo paga por lo que usa. Considere la posibilidad de utilizar un clúster sin servidor si sus aplicaciones necesitan una capacidad de streaming bajo demanda que se escale y reduzca verticalmente de forma automática.

MSK sin servidor es totalmente compatible con Apache Kafka, por lo que puede usar cualquier aplicación cliente compatible para producir y consumir datos. También se integra con los siguientes servicios:

- AWS PrivateLink para proporcionar conectividad privada
- AWS Identity and Access Management (IAM) para la autenticación y la autorización mediante lenguajes Java y no Java. Para obtener instrucciones sobre la configuración de clientes para IAM, consulte Configuración de clientes para el control de acceso de IAM.
- AWS Glue Registro de esquemas para la administración de esquemas
- Amazon Managed Service para Apache Flink para el procesamiento de flujos basados en Apache Flink
- AWS Lambda para el procesamiento de eventos

> **Note**
>
> MSK sin servidor requiere el control de acceso de IAM para todos los clústeres. No se admiten las listas de control de acceso de Apache Kafka (ACLs). Para obtener más información, consulte the section called “Control de acceso de IAM”.

Para obtener información sobre las cuotas de servicio que se aplican a MSK sin servidor, consulte the section called “Cuota para clústeres sin servidor”.

Para ayudarle a empezar con los clústeres sin servidor y para obtener más información sobre las opciones de configuración y supervisión de los clústeres sin servidor, consulte lo siguiente.

Temas

- Uso de clústeres sin servidor de MSK
- Propiedades de configuración para los clústeres sin servidor de MSK
- Supervisión de los clústeres sin servidor de MSK

Uso de clústeres sin servidor de MSK

En este tutorial, se muestra un ejemplo de cómo puede crear un clúster sin servidor de MSK, crear un equipo cliente al que pueda acceder y utilizar el cliente para crear temas en el clúster y escribir datos en esos temas. Este ejercicio no representa todas las opciones que puede elegir al crear un clúster sin servidor. En diferentes partes de este ejercicio, elegimos opciones predeterminadas por motivos de simplicidad. Esto no significa que sean las únicas opciones que funcionan para configurar un clúster sin servidor. También puedes usar la API AWS CLI o la API de Amazon MSK. Para más información, consulte la referencia de la API de Amazon MSK 2.0.

Temas

- Creación de un clúster sin servidor de MSK
- Creación de un rol de IAM para los temas del clúster sin servidor de MSK
- Creación de un equipo cliente para acceder al clúster sin servidor de MSK
- Creación de un tema de Apache Kafka
- Producción y consumo de datos en MSK sin servidor
- Eliminación de los recursos que creó para MSK sin servidor

Creación de un clúster sin servidor de MSK

En este paso, se realizan dos tareas. En primer lugar, debe crear un clúster sin servidor de MSK con la configuración predeterminada. En segundo lugar, debe recopilar información sobre el clúster. Esta

es la información que necesitará en los pasos posteriores al crear un cliente que pueda enviar datos al clúster.

Creación de un clúster sin servidor

1. Inicia sesión en la AWS Management Console consola Amazon MSK y ábrela en https://console.aws.amazon.com/msk/casa.
2. Elija Create cluster.
3. En Método de creación, deje seleccionada la opción Creación rápida. La opción Creación rápida le permite crear un clúster sin servidor con la configuración predeterminada.
4. En Nombre de clúster, escriba un nombre descriptivo, como **`msk-serverless-tutorial-cluster`**.
5. En Propiedades generales del clúster, elija Sin servidor como Tipo de clúster. Utilice los valores predeterminados para el resto de las propiedades generales del clúster.
6. Observe la tabla que figura bajo Todas las configuraciones del clúster. En esta tabla se muestran los valores predeterminados de las configuraciones importantes, como las redes y la disponibilidad, e indica si puede cambiar cada configuración después de crear el clúster. Para cambiar una configuración antes de crear el clúster, debe elegir la opción Creación personalizada en Método de creación.

   > **Note**
   >
   > Puede conectar clientes de hasta cinco tipos diferentes VPCs con clústeres de MSK Serverless. Para ayudar a las aplicaciones cliente a cambiar a otra zona de disponibilidad en caso de una interrupción, debe especificar al menos dos subredes en cada VPC.

7. Elija Create cluster.

Recopilación de información sobre el clúster

1. En la sección Resumen del clúster, elija Ver información del cliente. Este botón permanece atenuado hasta que Amazon MSK termina de crear el clúster. Es posible que tenga que esperar unos minutos hasta que el botón se active para poder utilizarlo.
2. Copie la cadena bajo la etiqueta Punto de conexión. Esta es la cadena del servidor de arranque.
3. Elija la pestaña Propiedades.

4. En la sección Configuración de red, copie las subredes y el grupo IDs de seguridad y guárdelos, ya que necesitará esta información más adelante para crear una máquina cliente.
5. Elija cualquiera de las subredes. Esto abre la consola de Amazon VPC. Encuentre el ID de la VPC de Amazon VPC asociada a la subred. Guarde este ID de la VPC de Amazon VPC para un uso posterior.

Paso siguiente

Creación de un rol de IAM para los temas del clúster sin servidor de MSK

Creación de un rol de IAM para los temas del clúster sin servidor de MSK

En este paso, se realizan dos tareas. La primera tarea consiste en crear una política de IAM que conceda acceso para crear temas en el clúster y enviarles datos. La segunda tarea consiste en crear un rol de IAM y asociarle esta política. En un paso posterior, se crea un equipo cliente que asume este rol y lo utiliza para crear un tema en el clúster y enviarle datos.

Creación de una política de IAM que permita crear temas y escribir en ellos

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, seleccione Políticas.
3. Elija Crear política.
4. Seleccione la pestaña JSON y, a continuación, sustituya el JSON de la ventana del editor por el siguiente JSON.

   En el ejemplo siguiente, sustituya lo siguiente:

   - *region*con el código del Región de AWS lugar en el que creó el clúster.
   - *Account-ID*con tu Cuenta de AWS ID.
   - *msk-serverless-tutorial-cluster*/*c07c74ea-5146-4a03-add1-9baa787a5b14-s3*y *msk-serverless-tutorial-cluster* con su ID de clúster sin servidor y el nombre del tema.

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

```
      "Effect": "Allow",
      "Action": [
        "kafka-cluster:Connect",
        "kafka-cluster:DescribeCluster"
      ],
      "Resource": [
        "arn:aws:kafka:region:Account-ID:cluster/msk-serverless-tutorial-
cluster/c07c74ea-5146-4a03-add1-9baa787a5b14-s3"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kafka-cluster:CreateTopic",
        "kafka-cluster:WriteData",
        "kafka-cluster:DescribeTopic"
      ],
      "Resource": [
        "arn:aws:kafka:region:Account-ID:topic/msk-serverless-tutorial-cluster/*"
      ]
    }
  ]
}
```

Para obtener instrucciones sobre cómo escribir políticas de seguridad, consultethe section called "Control de acceso de IAM".

5. Elija Siguiente: Etiquetas.
6. Elija Siguiente: Revisar.
7. En Nombre de política, escriba un nombre descriptivo, como **`msk-serverless-tutorial-policy`**.
8. Elija Crear política.

Creación de un rol de IAM y asociarle la política

1. En el panel de navegación, seleccione Roles.
2. Seleccione Crear rol.
3. En Casos de uso comunes, elija EC2y, a continuación, elija Siguiente: permisos.
4. En el cuadro de búsqueda, escriba el nombre de la política que creó anteriormente para este tutorial. A continuación, seleccione la casilla situada a la izquierda de la política.

5. Elija Siguiente: Etiquetas.
6. Elija Siguiente: Revisar.
7. En Nombre de rol, escriba un nombre descriptivo, como **`msk-serverless-tutorial-role`**.
8. Seleccione Crear rol.

Paso siguiente

Creación de un equipo cliente para acceder al clúster sin servidor de MSK

Creación de un equipo cliente para acceder al clúster sin servidor de MSK

En este paso, se realizan dos tareas. La primera tarea consiste en crear una EC2 instancia de Amazon para utilizarla como máquina cliente de Apache Kafka. La segunda tarea consiste en instalar las herramientas de Java y Apache Kafka en el equipo.

Creación de un equipo cliente

1. Abre la EC2 consola de Amazon en https://console.aws.amazon.com/ec2/.
2. Seleccione Iniciar instancia.
3. Escriba un Nombre descriptivo para el equipo cliente, como **`msk-serverless-tutorial-client`**.
4. Deje seleccionado Amazon Linux 2 AMI (HVM): Kernel 5.10, tipo de volumen SSD en Tipo de imagen de máquina de Amazon (AMI).
5. Deje seleccionado el tipo de instancia t2.micro.
6. En Par de claves (inicio de sesión), seleccione Crear un nuevo par de claves. Introduzca **`MSKServerlessKeyPair`** en Nombre de par de claves. Luego, elija Descargar par de claves. También puede utilizar un par de claves existente.
7. En Configuración de red, elija Editar.
8. En VPC, escriba el ID de la nube privada virtual (VPC) del clúster sin servidor. Esta es la VPC según el servicio de Amazon VPC, cuyo ID guardó después de crear el clúster.
9. En Subred, elija la subred cuyo ID guardó después de crear el clúster.
10. En Firewall (grupos de seguridad), seleccione el grupo de seguridad asociado al clúster. Este valor funciona si ese grupo de seguridad tiene una regla de entrada que permite el tráfico desde el grupo de seguridad hacia sí mismo. Con esta regla, los miembros del mismo grupo de

seguridad se pueden comunicar entre sí. Para más información, consulte Reglas del grupo de seguridad en la Guía para desarrolladores de Amazon VPC.

11. Amplíe la sección Detalles avanzados y elija el rol de IAM que creó en Creación de un rol de IAM para los temas del clúster sin servidor de MSK.
12. Elija Iniciar.
13. En el panel de navegación izquierdo, elija instancias. A continuación, selecciona la casilla de verificación de la fila que representa la EC2 instancia de Amazon recién creada. A partir de ahora, esta instancia se denominará equipo cliente.
14. Elija Conectar y siga las instrucciones para conectarse al equipo cliente.

Configuración de las herramientas de cliente de Apache Kafka en el equipo cliente

1. Para instalar Java, ejecute el siguiente comando en el equipo cliente:

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

2. Para obtener las herramientas de Apache Kafka que se necesitan para crear temas y enviar datos, ejecute los siguientes comandos:

```
wget https://archive.apache.org/dist/kafka/2.8.1/kafka_2.12-2.8.1.tgz
```

```
tar -xzf kafka_2.12-2.8.1.tgz
```

> **Note**
>
> Tras extraer el archivo de Kafka, asegúrate de que los scripts del `bin` directorio tengan los permisos de ejecución adecuados. Para ello, ejecute el siguiente comando.
>
> ```
> chmod +x kafka_2.12-2.8.1/bin/.sh
> ```

3. Vaya al directorio `kafka_2.12-2.8.1/libs` y ejecute el siguiente comando para descargar el archivo JAR de IAM de Amazon MSK. El JAR de IAM de Amazon MSK permite que el equipo cliente acceda al clúster.

```
wget https://github.com/aws/aws-msk-iam-auth/releases/download/v2.3.0/aws-msk-iam-auth-2.3.0-all.jar
```

Con este comando, también puede descargar otras versiones o versiones más recientes del archivo JAR de Amazon MSK IAM.

4. Vaya al directorio `kafka_2.12-2.8.1/bin`. Copie las siguientes configuraciones de propiedades y péguelas en un archivo nuevo. Asigne el nombre `client.properties` al archivo y guárdelo.

```
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
```

Paso siguiente

Creación de un tema de Apache Kafka

Creación de un tema de Apache Kafka

En este paso, utilizará el equipo cliente creado anteriormente para crear un tema en el clúster sin servidor.

Temas

- Configuración del entorno para la creación de temas
- Crear un tema y escribir datos en él

Configuración del entorno para la creación de temas

- Antes de crear un tema, asegúrese de haber descargado el archivo JAR de IAM de AWS MSK en el directorio de la instalación de Kafka. `libs/` Si aún no lo ha hecho, ejecute el siguiente comando en el directorio de Kafka. `libs/`

```
wget https://github.com/aws/aws-msk-iam-auth/releases/download/v2.3.0/aws-msk-iam-auth-2.3.0-all.jar
```

  Este archivo JAR es necesario para la autenticación de IAM con su clúster MSK Serverless.
- Al ejecutar los comandos de Kafka, es posible que tengas que asegurarte de que `classpath` incluyen el archivo JAR de IAM de AWS MSK. Para ello, siga uno de estos pasos:

- Configure la variable de `CLASSPATH` entorno para incluir las bibliotecas de Kafka, como se muestra en el siguiente ejemplo.

```
export CLASSPATH=<path-to-your-kafka-installation>/libs/*:<path-to-your-kafka-installation>/libs/aws-msk-iam-auth-2.3.0-all.jar
```

- Ejecute los comandos de Kafka utilizando el comando Java completo con explícito`classpath`, como se muestra en el siguiente ejemplo.

```
java -cp "<path-to-your-kafka-installation>/libs/*:<path-to-your-kafka-installation>/libs/aws-msk-iam-auth-2.3.0-all.jar"
 org.apache.kafka.tools.TopicCommand --bootstrap-server $BS --command-config
 client.properties --create --topic msk-serverless-tutorial --partitions 6
```

Crear un tema y escribir datos en él

1. En el siguiente `export` comando, *`my-endpoint`* sustitúyala por la cadena bootstrap-server que guardaste después de crear el clúster. A continuación, vaya al directorio `kafka_2.12-2.8.1/bin` del equipo cliente y ejecute el comando `export`.

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

2. Ejecute el siguiente comando para crear un tema denominado `msk-serverless-tutorial`.

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

Paso siguiente

Producción y consumo de datos en MSK sin servidor

Producción y consumo de datos en MSK sin servidor

En este paso, se producirán y consumirán datos mediante el tema que creó en el paso anterior.

Producción y consumo de mensajes

1. Ejecute el siguiente comando para crear un productor de la consola.

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

2. Escriba el mensaje que desee y pulse Enter (Entrar). Repita este paso dos o tres veces. Cada vez que introduzca una línea y pulse Entrar, dicha línea se enviará al clúster como un mensaje separado.
3. Mantenga abierta la conexión al equipo cliente y, a continuación, abra una segunda conexión independiente a dicho equipo en una nueva ventana.
4. Utilice la segunda conexión a la máquina cliente para crear un consumidor de consola con el siguiente comando. *my-endpoint*Sustitúyala por la cadena del servidor de arranque que guardaste después de crear el clúster.

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

   Comenzará a ver los mensajes que introdujo anteriormente cuando utilizó el comando del productor de la consola.
5. Escriba más mensajes en la ventana del productor y observe cómo aparecen en la ventana del consumidor.

Si tiene `classpath` problemas al ejecutar estos comandos, asegúrese de ejecutarlos desde el directorio correcto. Además, asegúrese de que el AWS MSK IAM JAR esté en el `libs` directorio. Como alternativa, puede ejecutar los comandos de Kafka utilizando el comando Java completo con explicito`classpath`, como se muestra en el siguiente ejemplo.

```
java -cp "kafka_2.12-2.8.1/libs/*:kafka_2.12-2.8.1/libs/aws-msk-iam-auth-2.3.0-
all.jar" org.apache.kafka.tools.ConsoleProducer –broker-list $BS –producer.config
 client.properties –topic msk-serverless-tutorial
```

Paso siguiente

Eliminación de los recursos que creó para MSK sin servidor

Eliminación de los recursos que creó para MSK sin servidor

En este paso, eliminará los recursos que creó en este tutorial.

Eliminación del clúster

1. Abre la consola Amazon MSK en https://console.aws.amazon.com/msk/casa.
2. En la lista de clústeres, elija el clúster que creó para este tutorial.
3. En Acciones, elija Eliminar clúster.
4. Introduzca `delete` en el campo y, luego, elija Eliminar.

Detención del equipo cliente

1. Abre la EC2 consola de Amazon en https://console.aws.amazon.com/ec2/.
2. En la lista de EC2 instancias de Amazon, elige la máquina cliente que creaste para este tutorial.
3. Elija Estado de la instancia y, luego, Terminar instancia.
4. Elija Finalizar.

Eliminación de la política y el rol de IAM

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, seleccione Roles.
3. En el cuadro de búsqueda, escriba el nombre del rol de IAM que creó para este tutorial.
4. Elija el rol. A continuación, elija Eliminar rol para confirmar la eliminación.
5. En el panel de navegación, seleccione Políticas.
6. En el cuadro de búsqueda, escriba el nombre de la política que creó para este tutorial.
7. Elija la política para abrir su página de resumen. En la página Resumen de la política, elija Eliminar política.
8. Elija Eliminar.

Propiedades de configuración para los clústeres sin servidor de MSK

Amazon MSK establece las propiedades de configuración de los agentes para los clústeres sin servidor. No puede cambiar los ajustes de las propiedades de configuración de estos agentes. Sin embargo, puede establecer o modificar las siguientes propiedades de configuración a nivel del tema. El resto de las propiedades de configuración a nivel del tema no son configurables.

Propiedad de configuración	Predeterminado/a	Editable	Valor máximo permitido
cleanup.policy	Eliminar	Sí, pero solo en el momento de crear el tema	
compression.type	Productor	Sí	
max.message.bytes	1048588	Sí	8388608 (8 MiB)
message.timestamp.difference.max.ms	long.max	Sí	
message.timestamp.type	CreateTime	Sí	
retention.bytes	250 GiB	Sí	Ilimitado; configúrelo en -1 para una retención ilimitada
retention.ms	7 días	Sí	Ilimitado; configúrelo en -1 para una retención ilimitada

Para configurar o modificar estas propiedades de configuración a nivel del tema, puede utilizar herramientas de línea de comandos de Apache Kafka. Consulte 3.2 Topic-level Configs en la documentación oficial de Apache Kafka para obtener más información y ejemplos de cómo establecerlas.

> **Note**
>
> No puedes modificar la configuración de segment.bytes para los temas de MSK Serverless. Sin embargo, una aplicación de Kafka Streams podría intentar crear un tema interno con un valor de configuración segment.bytes diferente del que permite MSK Serverless. Para obtener información sobre la configuración de Kafka Streams con MSK Serverless, consulte. Uso de Kafka Streams con los corredores de MSK Express y MSK Serverless

Cuando utilice las herramientas de línea de comandos de Apache Kafka con Amazon MSK sin servidor, asegúrese de haber completado los pasos 1 a 4 de la sección To set up Apache Kafka client tools on the client machine de la documentación de introducción a Amazon MSK sin servidor. Además, debe incluir el parámetro `--command-config client.properties` en sus comandos.

Por ejemplo, se puede utilizar el siguiente comando para modificar la propiedad de configuración del tema retention.bytes y establecer una retención ilimitada:

```
<path-to-your-kafka-client-installation>/bin/kafka-configs.sh –bootstrap-
server <bootstrap_server_string> –command-config client.properties --entity-type topics
 --entity-name <topic_name> --alter --add-config retention.bytes=-1
```

En este ejemplo, *`<bootstrap server string>`* sustitúyalo por el punto final del servidor bootstrap de tu clúster Amazon MSK Serverless y *`<topic_name>`* por el nombre del tema que deseas modificar.

El parámetro `--command-config client.properties` garantiza que la herramienta de línea de comandos de Kafka utilice los ajustes de configuración adecuados para comunicarse con el clúster de Amazon MSK sin servidor.

Supervisión de los clústeres sin servidor de MSK

Amazon MSK se integra con Amazon CloudWatch para que pueda recopilar, ver y analizar las métricas de su clúster MSK Serverless. Las métricas que se muestran en la tabla siguiente están disponibles para todos los clústeres sin servidor. Como estas métricas se publican como puntos de datos individuales para cada partición del tema, recomendamos visualizarlas como una estadística “SUM” para obtener una vista por tema.

Amazon MSK publica `PerSec` las métricas con CloudWatch una frecuencia de una vez por minuto. Esto significa que la estadística “SUM” de un periodo de un minuto representa con precisión los datos por segundo de las métricas `PerSec`. Para recopilar datos por segundo durante un período superior a un minuto, utilice la siguiente expresión CloudWatch matemática:. `m1 * 60/PERIOD(m1)`

Métricas disponibles en el nivel de supervisión DEFAULT

Nombre	Cuando está visible	Dimensiones	Descripción
`BytesInPerSec`	Después de que un productor escribe en un tema	Nombre del clúster, tema	El número de bytes por segundo recibidos de los clientes. Esta métrica está disponible para cada tema.
`BytesOutPerSec`	Después de que un grupo de consumidores consume de un tema	Nombre del clúster, tema	El número de bytes por segundo enviados a los clientes. Esta métrica está disponible para cada tema.
`FetchMessageConversionsPerSec`	Después de que un grupo de consumidores consume de un tema	Nombre del clúster, tema	El número de conversiones de mensajes de recuperación por segundo para el tema.
`EstimatedMaxTimeLag`	Después de que un grupo de consumidores consume de un tema	Nombre del clúster, grupo de consumidores, tema	Una estimación temporal de la MaxOffsetLag métrica.
`MaxOffsetLag`	Después de que un grupo de consumidores consume de un tema	Nombre del clúster, grupo de consumidores, tema	El retraso máximo de desplazamiento en todas las particiones de un tema.
`MessagesInPerSec`	Después de que un productor	Nombre del clúster, tema	El número de mensajes entrantes por segundo para el tema.

Nombre	Cuando está visible	Dimensiones	Descripción
	escribe en un tema		
`ProduceMessageConversionsPerSec`	Después de que un productor escribe en un tema	Nombre del clúster, tema	El número de conversiones de mensajes de producción por segundo para el tema.
`SumOffsetLag`	Después de que un grupo de consumidores consume de un tema	Nombre del clúster, grupo de consumidores, tema	El retraso de desplazamiento agregado de todas las particiones de un tema.

Visualización de las métricas de MSK sin servidor

1. Inicie sesión en AWS Management Console y abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, en Métricas y, luego, Todas las métricas.
3. En las métricas, busque el término **`kafka`**.
4. Elija AWS, Kafka, Nombre del clúster o Tema o AWS, Kafka, Nombre del clúster, Grupo de consumidores o Tema para ver diferentes métricas.

Descripción de MSK Connect

MSK Connect es una característica de Amazon MSK que facilita a los desarrolladores la transmisión de datos hacia y desde sus clústeres de Apache Kafka. MSK Connect utiliza las versiones 2.7.1 o 3.7.x de Kafka Connect, que son marcos de código abierto para conectar los clústeres de Apache Kafka con sistemas externos, como bases de datos, índices de búsqueda y sistemas de archivos. Con MSK Connect, puede implementar conectores totalmente administrados diseñados para Kafka Connect que mueven o extraen datos de almacenes de datos populares, como Amazon S3 y Amazon OpenSearch Service. Puede implementar conectores desarrollados por terceros, como Debezium, para transmitir los registros de cambios de las bases de datos a un clúster de Apache Kafka, o implementar un conector existente sin cambios de código. Los conectores se escalan automáticamente para adaptarse a los cambios de carga y solo pagará por los recursos que utilice.

Utilice los conectores de origen para importar datos de sistemas externos a sus temas. Con los conectores de recepción, puede exportar datos de sus temas a sistemas externos.

MSK Connect admite conectores para cualquier clúster de Apache Kafka con conectividad a una Amazon VPC, ya sea un clúster de MSK o un clúster de Apache Kafka alojado de forma independiente.

MSK Connect monitorea continuamente el buen estado general y el estado de entrega de los conectores, parchea y administra el hardware subyacente y escala automáticamente los conectores para adaptarlos a los cambios en el rendimiento.

Para comenzar a utilizar MSK Connect, consulte the section called “Introducción”.

Para obtener información sobre los AWS recursos que puede crear con MSK Connect, consulte the section called “Descripción de los conectores”the section called “Creación de complementos personalizados”, ythe section called “Descripción de los procesos de trabajo de MSK Connect”.

Para obtener información sobre la API de MSK Connect, consulte la Referencia de la API de Amazon MSK Connect.

Beneficios de usar Amazon MSK Connect

Apache Kafka es una de las plataformas de transmisión de código abierto más utilizadas para la ingesta y el procesamiento de flujos de datos en tiempo real. Con Apache Kafka, puede desacoplar y escalar de forma independiente las aplicaciones que producen y consumen datos.

Kafka Connect es un componente importante para la creación y la ejecución de aplicaciones de transmisión con Apache Kafka. Kafka Connect proporciona una forma estandarizada de trasladar datos entre Kafka y los sistemas externos. Kafka Connect es altamente escalable y puede administrar grandes volúmenes de datos. Kafka Connect proporciona un potente conjunto de operaciones y herramientas de API para configurar, implementar y supervisar conectores que trasladan datos entre los temas de Kafka y sistemas externos. Con estas herramientas, puede personalizar y ampliar la funcionalidad de Kafka Connect para satisfacer las necesidades específicas de su aplicación de transmisión.

Es posible que encuentre dificultades cuando utilice clústeres de Apache Kafka Connect por sí solos o cuando intente migrar aplicaciones de código abierto de Apache Kafka Connect a AWS. Estos desafíos incluyen el tiempo necesario para configurar la infraestructura e implementar las aplicaciones, los obstáculos de ingeniería al configurar los clústeres de Apache Kafka Connect autoadministrados y la sobrecarga operativa administrativa.

Para hacer frente a estos desafíos, recomendamos que utilice Amazon Managed Streaming para Apache Kafka Connect (Amazon MSK Connect) para migrar sus aplicaciones de código abierto Apache Kafka Connect a AWS. Amazon MSK Connect simplifica el uso de Kafka Connect para transmitir datos entre los clústeres de Apache Kafka y sistemas externos, como bases de datos, índices de búsqueda y sistemas de archivos.

Estas son algunas de las ventajas de migrar a Amazon MSK Connect:

- Eliminación de la sobrecarga operativa: Amazon MSK Connect elimina la carga operativa asociada a la aplicación de parches, el aprovisionamiento y el escalado de los clústeres de Apache Kafka Connect. Amazon MSK Connect supervisa de forma continua el estado de los clústeres de Connect y automatiza la aplicación de parches y las actualizaciones de versiones sin interrumpir las cargas de trabajo.
- Reinicio automático de las tareas de Connect: Amazon MSK Connect puede recuperar automáticamente las tareas fallidas para reducir las interrupciones en la producción. Los errores en las tareas pueden ser el resultado de errores temporales, como superar el límite de conexiones TCP en Kafka o el reequilibrio de tareas cuando se incorporan nuevos procesos de trabajo al grupo de consumidores de conectores de recepción.
- Escalado vertical y horizontal automático: Amazon MSK Connect permite que el conector de aplicaciones se escale automáticamente para que sea compatible con mayores rendimientos. Amazon MSK Connect administra el escalado por usted. Solo necesita especificar el número de procesos de trabajo en el grupo de escalado automático y los umbrales de utilización. Puede

utilizar la operación de la `UpdateConnector` API de Amazon MSK Connect para escalar o reducir verticalmente la v CPUs entre 1 y 8, de manera que se CPUs ajusten al rendimiento variable.

- Conectividad de red privada: Amazon MSK Connect se conecta de forma privada a los sistemas de origen y recepción a través AWS PrivateLink de los nombres DNS privados.

Introducción a MSK Connect

Este es un step-by-step tutorial que utiliza la AWS Management Console para crear un clúster de MSK y un conector de recepción que envía los datos del clúster a un bucket de S3.

Temas

- Configuración de los recursos necesarios para MSK Connect
- Creación de un complemento personalizado
- Creación de un equipo cliente y un tema de Apache Kafka
- Crea conector
- Envío de datos al clúster de MSK

Configuración de los recursos necesarios para MSK Connect

En este paso, creará los siguientes recursos que necesitará para este escenario de introducción:

- Un bucket de Amazon S3 que sirva de destino para recibir los datos del conector.
- Un clúster de MSK al que enviará los datos. A continuación, el conector leerá los datos de este clúster y los enviará al bucket de S3 de destino.
- Política de IAM que contiene los permisos para escribir en el bucket de S3 de destino.
- Un rol de IAM que permite al conector escribir en el bucket de S3 de destino. Agregará la política de IAM que creó a este rol.
- Un punto de conexión de VPC de Amazon VPC para poder enviar datos desde la Amazon VPC que tiene el clúster y el conector a Amazon S3.

Creación del bucket de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.

2. Elija Crear bucket.
3. Para el nombre del bucket, introduzca un nombre descriptivo, como `amzn-s3-demo-bucket-mkc-tutorial`.
4. Desplácese hacia abajo y seleccione Crear bucket.
5. En la lista de buckets, elija el bucket que acaba de crear.
6. Elija Crear carpeta.
7. Introduzca `tutorial` para el nombre de la carpeta, desplácese hacia abajo y elija Crear carpeta.

Creación del clúster

1. ¿Abrir la consola de Amazon MSK en https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. En el panel izquierdo, en Clústeres de MSK, seleccione Clústeres.
3. Elija Create cluster.
4. En Método de creación, elija Creación personalizada.
5. Para el nombre del clúster, ingrese **`mkc-tutorial-cluster`**.
6. En Tipo de clúster, selecciona Aprovisionado.
7. Elija Siguiente.
8. En Redes, elija una Amazon VPC. A continuación, seleccione las zonas de disponibilidad y las subredes que desee utilizar. Recuerde IDs la Amazon VPC y las subredes que seleccionó porque las necesitará más adelante en este tutorial.
9. Elija Siguiente.
10. En Métodos de control de acceso, asegúrese de seleccionar solo Acceso no autenticado.
11. En Cifrado, asegúrese de seleccionar solo Texto sin formato.
12. Continúe con el asistente y, a continuación, seleccione Crear clúster. Esto le lleva a la página Detalles del clúster. En esa página, en Grupos de seguridad aplicados, busque el ID del grupo de seguridad. Recuerde ese ID, ya que lo necesitará más tarde en este tutorial.

Creación de una política de IAM con permisos para escribir en el bucket de S3

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, seleccione Políticas.

3. Elija Crear política.
4. En el editor de políticas, elija JSON y, a continuación, sustituya el JSON de la ventana del editor por el siguiente JSON.

   En el ejemplo siguiente, *<amzn-s3-demo-bucket-my-tutorial>* reemplace por el nombre del bucket de S3.

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

   Para obtener instrucciones acerca de cómo escribir políticas de seguridad, consultethe section called “Control de acceso de IAM”.
5. Elija Siguiente.
6. En la página Revisar y crear, haga lo siguiente:

   a. En Nombre de política, ingrese un nombre descriptivo, como**mkc-tutorial-policy**.

b. En Permisos definidos en esta política, revise o edite los permisos definidos en su política.

c. (Opcional) Para ayudar a identificar, organizar o buscar la política, elija Agregar nueva etiqueta para agregar etiquetas como pares clave-valor. Por ejemplo, agrega una etiqueta a tu política con el par clave-valor de y. **`Environment Test`**

   Para obtener más información sobre el uso de etiquetas, consulte Etiquetas para AWS Identity and Access Management recursos en la Guía del usuario de IAM.

7. Elija Crear política.

Creación del rol de IAM que pueda escribir en el bucket de destino

1. En el panel de navegación de la consola de IAM, elija Funciones y, a continuación, elija Crear función.
2. En la página Seleccionar entidad de confianza, haga lo siguiente:

   a. En Tipo de entidad de confianza, elija Servicio de AWS.

   b. Para el servicio o el caso de uso, elija S3.

   c. En Caso de uso, selecciona S3.

3. Elija Siguiente.
4. En la página Agregar permisos, haga lo siguiente:

   a. En el cuadro de búsqueda de Políticas de permisos, escriba el nombre de la política que creó anteriormente para este tutorial. Por ejemplo, mkc-tutorial-policy. A continuación, elija la casilla situada a la izquierda del nombre de la política.

   b. (Opcional) Configure un límite de permisos. Se trata de una característica avanzada que está disponible para los roles de servicio, pero no para los roles vinculados a servicios. Para obtener información sobre cómo establecer un límite de permisos, consulte Crear funciones y adjuntar políticas (consola) en la Guía del usuario de IAM.

5. Elija Siguiente.
6. En la página Nombrar, revisar y crear, haga lo siguiente:

   a. En Nombre de rol, ingrese un nombre descriptivo, como**`mkc-tutorial-role`**.

   > **Important**
   >
   > Cuando asigne un nombre a un rol, tenga en cuenta lo siguiente:

- Los nombres de rol deben ser únicos dentro de su Cuenta de AWS, y no se puedesn hacer únicos mediante mayúsculas y minúsculas.

  Por ejemplo, no puedes crear roles denominados tanto **`PRODROLE`** como **`prodrole`**. Cuando se utiliza un nombre de rol en una política o como parte de un ARN, el nombre de rol distingue entre mayúsculas y minúsculas, sin embargo, cuando un nombre de rol les aparece a los clientes en la consola, como por ejemplo durante el proceso de inicio de sesión, el nombre de rol no distingue entre mayúsculas y minúsculas.
- Dado que otras entidades podrían hacer referencia al rol, no es posible editar el nombre del rol una vez creado.

b. (Opcional) En Descripción, ingrese una descripción para el rol.

c. (Opcional) Para editar los casos de uso y los permisos del rol, en el paso 1: seleccionar entidades de confianza o en el paso 2: agregar secciones de permisos, seleccione Editar.

d. (Opcional) Para ayudar a identificar, organizar o buscar el rol, elija Agregar nueva etiqueta para agregar etiquetas como pares clave-valor. Por ejemplo, agrega una etiqueta a tu rol con el par clave-valor de y. **`ProductManager John`**

   Para obtener más información sobre el uso de etiquetas, consulte Etiquetas para AWS Identity and Access Management recursos en la Guía del usuario de IAM.

7. Revise el rol y, a continuación, elija Crear rol.

Autorización para que MSK Connect asuma el rol

1. En la consola de IAM, en el panel izquierdo, en Administración de acceso, seleccione Roles.
2. Busque el `mkc-tutorial-role` y selecciónelo.
3. En el Resumen del rol, elija la pestaña Relaciones de confianza.
4. Elija Editar relación de confianza.
5. Reemplace la política existente por el siguiente JSON.

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

```
      "Principal": {
        "Service": "kafkaconnect.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

6. Elija Actualizar política de confianza.

Creación de un punto de conexión de VPC de Amazon VPC desde la VPC del clúster a Amazon S3

1. Abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc/.
2. En el panel izquierdo, elija Puntos de conexión.
3. Elija Crear punto de conexión.
4. En Nombre del servicio, elija el servicio com.amazonaws.us-east-1.s3 y el tipo de puerta de enlace.
5. Elija la VPC del clúster y, a continuación, seleccione la casilla situada a la izquierda de la tabla de enrutamiento que está asociada a las subredes del clúster.
6. Elija Crear punto de conexión.

Paso siguiente

Creación de un complemento personalizado

Creación de un complemento personalizado

Un complemento contiene el código que define la lógica del conector. En este paso, creará un complemento personalizado que contenga el código del conector de recepción de Amazon S3 de Lenses. En un paso posterior, cuando cree el conector de MSK, especifique que su código está en este complemento personalizado. Puede usar el mismo complemento para crear varios conectores de MSK con diferentes configuraciones.

Creación del complemento personalizado

1. Descargue el conector de S3.

2. Cargue el archivo ZIP en un bucket de S3 al que tenga acceso. Para obtener información sobre cómo cargar archivos en Amazon S3, consulte Carga de objetos en la guía del usuario de Amazon S3.
3. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/.
4. En el panel izquierdo, expanda MSK Connect y, a continuación, seleccione Complementos personalizados.
5. Seleccione Crear complemento personalizado.
6. Elija Examinar S3.
7. En la lista de buckets, busque el bucket en el que ha cargado el archivo ZIP, y elija ese bucket.
8. En la lista de objetos del bucket, seleccione el botón de opción situado a la izquierda del archivo ZIP y, a continuación, seleccione el botón denominado Elegir.
9. Ingrese `mkc-tutorial-plugin` para el nombre del complemento personalizado y, a continuación, seleccione Crear complemento personalizado.

Puede que tarde varios AWS minutos en finalizar la creación del complemento personalizado. Cuando se complete el proceso de creación, verá el siguiente mensaje en un banner en la parte superior de la ventana del navegador.

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

Paso siguiente

Creación de un equipo cliente y un tema de Apache Kafka

Creación de un equipo cliente y un tema de Apache Kafka

En este paso, cree una EC2 instancia de Amazon para utilizarla como instancia de cliente de Apache Kafka. A continuación, utilice esta instancia para crear un tema en el clúster.

Creación de un equipo cliente

1. Abre la EC2 consola de Amazon en https://console.aws.amazon.com/ec2/.
2. Elija iniciar instancias.
3. Escriba un nombre para el equipo cliente, como **`mkc-tutorial-client`**.

4. Deje seleccionado Amazon Linux 2 AMI (HVM): Kernel 5.10, tipo de volumen SSD en Tipo de imagen de máquina de Amazon (AMI).
5. Elija el tipo de instancia t2.xlarge.
6. En Par de claves (inicio de sesión), seleccione Crear un nuevo par de claves. Introduzca **`mkc-tutorial-key-pair`** en Nombre del par de claves y, a continuación, seleccione Descargar par de claves. También puede utilizar un par de claves existente.
7. Seleccione Iniciar instancia.
8. Elija View Instances (Ver instancias). A continuación, en la columna Grupos de seguridad, elija el grupo de seguridad asociado a la nueva instancia. Copie el ID del grupo de seguridad y guárdelo para más adelante.

Concesión de permiso al cliente recién creado para que envíe datos al clúster

1. Abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc/.
2. En el panel izquierdo, en SEGURIDAD, elija Grupos de seguridad. En la columna ID de grupo de seguridad, busque el grupo de seguridad del clúster. Guardó el ID de este grupo de seguridad cuando creó el clúster en the section called “Configuración de los recursos necesarios para MSK Connect”. Seleccione este grupo de seguridad seleccionando la casilla situada a la izquierda de su fila. Asegúrese de que no haya otros grupos de seguridad seleccionados simultáneamente.
3. En la parte inferior de la pantalla, elija la pestaña Reglas de entrada.
4. Elija Editar reglas de entrada.
5. En la parte inferior izquierda de la pantalla, elija Añadir regla.
6. En la nueva regla, elija All traffic (Todo el tráfico) en la columna Type (Tipo). En el campo de la derecha de la columna Origen, escriba el ID del grupo de seguridad del equipo cliente. Este es el ID del grupo de seguridad que guardó después de crear el equipo cliente.
7. Seleccione Guardar reglas. Su clúster de MSK ahora aceptará todo el tráfico del cliente que creó en el procedimiento anterior.

Creación de un tema

1. Abre la EC2 consola de Amazon en https://console.aws.amazon.com/ec2/.
2. En la tabla de instancias, elija `mkc-tutorial-client`.
3. Cerca de la parte superior de la pantalla, seleccione Conectar y siga las instrucciones para conectarse a la instancia.

4. Instale Java en la instancia del cliente ejecutando el siguiente comando:

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

5. Ejecute el siguiente comando para descargar Apache Kafka.

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

> **Note**
>
> Si desea utilizar un sitio espejo que no sea el utilizado en este comando, puede elegir uno diferente en el sitio web de Apache.

6. Ejecute el siguiente comando en el directorio donde descargó el archivo TAR del paso anterior.

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

7. Vaya al directorio kafka_2.12-2.2.1.
8. ¿Abrir la consola de Amazon MSK en https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
9. En el panel izquierdo, seleccione Clústeres y, a continuación, elija el nombre `mkc-tutorial-cluster`.
10. Seleccione Ver información del cliente.
11. Copie la cadena de conexión de texto sin formato.
12. Seleccione Listo.
13. Ejecute el siguiente comando en la instancia del cliente (`mkc-tutorial-client`) y *`bootstrapServerString`* sustituya este por el valor que guardó al ver la información del cliente del clúster.

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

Si el comando se ejecuta correctamente, verá el siguiente mensaje: `Created topic mkc-tutorial-topic.`

Paso siguiente

Crea conector

Crea conector

En este procedimiento, se describe cómo crear un conector desde la AWS Management Console.

Creación del conector

1. ¿Iniciar sesión en la AWS Management Console y abrir la consola de Amazon MSK en https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. En el panel izquierdo, expanda MSK Connect y seleccione Conectores.
3. Elija Crear conector.
4. En la lista de complementos, seleccione `mkc-tutorial-plugin` y, a continuación, seleccione Siguiente.
5. Para el nombre del conector, ingrese `mkc-tutorial-connector`.
6. En la lista de clústeres, elija `mkc-tutorial-cluster`.
7. Copie la siguiente configuración y péguela en el campo de configuración del conector.

   Asegúrese de reemplazar la región por el código del lugar donde va a crear el conector. Región de AWS Además, en el ejemplo siguiente, reemplace el nombre del *<amzn-s3-demo-bucket-my-tutorial>* bucket de Amazon S3 por el nombre del bucket de Amazon S3 por el nombre del bucket.

   ```
   connector.class=io.confluent.connect.s3.S3SinkConnector
   s3.region=us-east-1
   format.class=io.confluent.connect.s3.format.json.JsonFormat
   flush.size=1
   schema.compatibility=NONE
   tasks.max=2
   topics=mkc-tutorial-topic
   partitioner.class=io.confluent.connect.storage.partitioner.DefaultPartitioner
   storage.class=io.confluent.connect.s3.storage.S3Storage
   s3.bucket.name=<amzn-s3-demo-bucket-my-tutorial>
   topics.dir=tutorial
   ```

8. En Permisos de acceso, elija `mkc-tutorial-role`.
9. Elija Siguiente. En la página Seguridad, vuelva a seleccionar Siguiente.

10. En la página Registros, seleccione Siguiente.
11. En Revisar y crear, elija Crear conector.

Paso siguiente

Envío de datos al clúster de MSK

Envío de datos al clúster de MSK

En este paso, envía los datos al tema de Apache Kafka que creó anteriormente y, a continuación, busca los mismos datos en el bucket de S3 de destino.

Envío de datos al clúster de MSK

1. En la carpeta `bin` de la instalación de Apache Kafka en la instancia del cliente, cree un archivo de texto denominado `client.properties` con el siguiente contenido.

```
security.protocol=SASL_SSL
sasl.mechanism=AWS_MSK_IAM
```

2. Ejecute el siguiente comando para crear un productor de la consola. *BootstrapBrokerString*Reemplace por el valor que obtuvo al ejecutar el comando anterior.

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

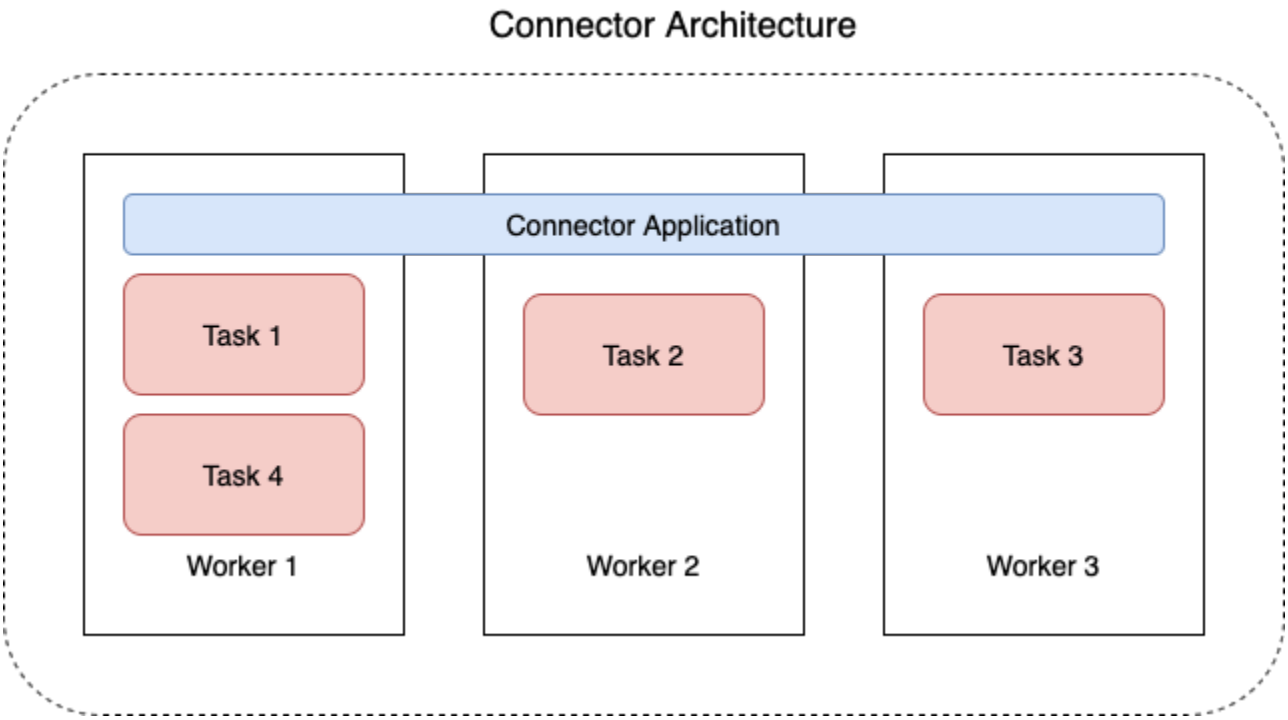
3. Escriba el mensaje que desee y pulse Enter (Entrar). Repita este paso dos o tres veces. Cada vez que introduzca una línea y pulse Enter (Entrar), dicha línea se envía al clúster de Apache Kafka como un mensaje separado.
4. Busque en el bucket de Amazon S3 de los mensajes que envió en el paso anterior.

Descripción de los conectores

Un conector integra sistemas externos y servicios de Amazon con Apache Kafka copiando continuamente los datos de streaming de un origen de datos a su clúster de Apache Kafka o copiando continuamente los datos de su clúster a un receptor de datos. Un conector también puede realizar una lógica ligera, como la transformación, la conversión de formato o el filtrado de datos antes de entregarlos a un destino. Los conectores de origen extraen datos de un origen de datos y

los envían al clúster, mientras que los conectores de destino extraen datos del clúster y los envían a un receptor de datos.

En el siguiente diagrama, se ilustra la arquitectura de un conector. Un proceso de trabajo es un proceso de máquina virtual Java (JVM) que ejecuta la lógica del conector. Cada proceso de trabajo crea un conjunto de tareas que se ejecutan en subprocesos paralelos y realizan el trabajo de copiar los datos. Las tareas no almacenan el estado y, por lo tanto, se pueden iniciar, detener o reiniciar en cualquier momento para proporcionar una canalización de datos flexible y escalable.



Descripción de la capacidad de los conectores

La capacidad total de un conector depende del número de trabajadores que tenga el conector, así como del número de unidades MSK Connect (MCUs) por trabajador. Cada MCU representa 1 vCPU de computación y 4 GiB de memoria. La memoria de la MCU corresponde a la memoria total de una instancia de proceso de trabajo y no a la memoria en montón que está en uso.

Los procesos de trabajo de MSK Connect consumen direcciones IP en las subredes proporcionadas por el cliente. Cada proceso de trabajo utiliza una dirección IP de una de las subredes proporcionadas por el cliente. Debe asegurarse de tener suficientes direcciones IP disponibles en las subredes proporcionadas en respuesta a una CreateConnector solicitud para tener en cuenta la capacidad especificada, especialmente al escalar automáticamente los conectores, donde la cantidad de trabajadores puede fluctuar.

Para crear un conector, debe elegir uno de los dos modos de capacidad siguientes.

- Aprovisionado: elija este modo si conoce los requisitos de capacidad del conector. Debe especificar dos valores:
  - El número de procesos de trabajo.
  - El número de por trabajador MCUs .
- Escalado automático: elija este modo si los requisitos de capacidad del conector son variables o si no los conoce de antemano. Cuando utiliza el modo de escalado automático, Amazon MSK Connect anula la propiedad `tasks.max` del conector con un valor que es proporcional al número de trabajadores que se ejecutan en el conector y al número MCUs de trabajadores por trabajador.

  Debe especificar tres conjuntos de valores:
  - El número mínimo y máximo de procesos de trabajo.
  - Los porcentajes de reducción vertical y escalado horizontal para el uso de la CPU, que se determinan mediante la métrica `CpuUtilization`. Cuando la métrica `CpuUtilization` del conector supera el porcentaje de escalado horizontal, MSK Connect aumenta la cantidad de procesos de trabajo que trabajan en el conector. Cuando la métrica `CpuUtilization` cae por debajo del porcentaje de reducción horizontal, MSK Connect reduce la cantidad de procesos de trabajo. El número de procesos de trabajo siempre se mantiene dentro de los números mínimo y máximo que se especificaron al crear el conector.
  - El número de por trabajador. MCUs

Para obtener más información acerca de los procesos de trabajo, consulte the section called “Descripción de los procesos de trabajo de MSK Connect”. Para obtener más información sobre las métricas de MSK Connect, consulte the section called “Monitorización”.

Crear un conector

En este procedimiento, se describe cómo crear un conector desde la AWS Management Console.

Crear un conector mediante el AWS Management Console

1. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/.
2. En el panel izquierdo, en MSK Connect, seleccione Conectores.
3. Elija Crear conector.

4. Puede elegir entre usar un complemento personalizado existente para crear el conector o crear primero un complemento personalizado nuevo. Para obtener información acerca de los complementos personalizados y cómo crearlos, consulte the section called “Creación de complementos personalizados”. En este procedimiento, supongamos que tiene un complemento personalizado que desea usar. En la lista de complementos personalizados, busque el que desee usar, seleccione la casilla situada a la izquierda y, a continuación, elija Siguiente.
5. Escriba un nombre y, opcionalmente, una descripción.
6. Haga clic en el clúster al que desea conectarse.
7. Especifique la configuración del conector. Los parámetros de configuración que debe especificar dependen del tipo de conector que desee crear. Sin embargo, algunos parámetros son comunes a todos los conectores, por ejemplo, los parámetros `connector.class` y `tasks.max`. A continuación, se muestra un ejemplo de configuración para el conector de recepción de Amazon S3 de Confluent.

```
connector.class=io.confluent.connect.s3.S3SinkConnector
tasks.max=2
topics=my-example-topic
s3.region=us-east-1
s3.bucket.name=amzn-s3-demo-bucket
flush.size=1
storage.class=io.confluent.connect.s3.storage.S3Storage
format.class=io.confluent.connect.s3.format.json.JsonFormat
partitioner.class=io.confluent.connect.storage.partitioner.DefaultPartitioner
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.kafka.connect.storage.StringConverter
schema.compatibility=NONE
```

8. A continuación, debe configurar la capacidad del conector. Puede elegir entre dos modos de capacidad: aprovisionada y con escalado automático. Para obtener información sobre estas dos opciones, consulte the section called “Descripción de la capacidad de los conectores”.
9. Elija la configuración de proceso de trabajo predeterminada o una configuración de proceso de trabajo personalizada. Para obtener información sobre la creación de configuraciones de procesos de trabajo personalizadas, consulte the section called “Descripción de los procesos de trabajo de MSK Connect”.
10. A continuación, especifique el rol de ejecución del servicio. Debe ser una función de IAM que MSK Connect pueda asumir y que conceda al conector todos los permisos que necesita para acceder a los recursos necesarios AWS . Estos permisos dependen de la lógica del conector.

Para obtener información acerca de cómo crear este rol, consulte the section called “Descripción del rol de ejecución de servicios”.

11. Seleccione Siguiente, revise la información de seguridad y, a continuación, vuelva a seleccionar Siguiente.
12. Especifique las opciones de registro que desee y elija Enviar. Para obtener más información acerca del registro, consulte the section called “Registro”.
13. Elija Crear conector.

Para usar la API MSK Connect para crear un conector, consulte CreateConnector.

Puede usar la `UpdateConnector` API para modificar la configuración del conector. Para obtener más información, consulte the section called “Actualizar un conector”.

Actualizar un conector

Este procedimiento describe cómo actualizar la configuración de un conector MSK Connect existente mediante el AWS Management Console.

Actualización de la configuración del conector mediante el AWS Management Console

1. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/.
2. En el panel izquierdo, en MSK Connect, seleccione Conectores.
3. Seleccione un conector existente.
4. Seleccione Editar la configuración del conector.
5. Actualice la configuración del conector. No puede anular el `connector.class` uso UpdateConnector. El siguiente ejemplo muestra un ejemplo de configuración para el conector Confluent Amazon S3 Sink.

```
connector.class=io.confluent.connect.s3.S3SinkConnector
tasks.max=2
topics=my-example-topic
s3.region=us-east-1
s3.bucket.name=amzn-s3-demo-bucket
flush.size=1
storage.class=io.confluent.connect.s3.storage.S3Storage
format.class=io.confluent.connect.s3.format.json.JsonFormat
partitioner.class=io.confluent.connect.storage.partitioner.DefaultPartitioner
```

```
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.kafka.connect.storage.StringConverter
schema.compatibility=NONE
```

6. Seleccione Submit (Enviar).
7. A continuación, puede supervisar el estado actual de la operación en la pestaña Operaciones del conector.

Para usar la API MSK Connect para actualizar la configuración de un conector, consulte UpdateConnector.

Conexión desde conectores

Las siguientes prácticas recomendadas pueden mejorar el rendimiento de la conectividad a Amazon MSK Connect.

No se superpongan IPs para el peering de Amazon VPC o Transit Gateway

Si utiliza la interconexión de Amazon VPC o Transit Gateway con Amazon MSK Connect, no configure el conector para llegar a los recursos de VPC interconectados que se encuentren dentro de los rangos de CIDR: IPs

- "10.99.0.0/16"
- "192.168.0.0/16"
- "172.21.0.0/16"

Creación de complementos personalizados

Un complemento es un AWS recurso que contiene el código que define la lógica del conector. Debe cargar un archivo JAR (o un archivo ZIP que contenga uno o más archivos JAR) en un bucket de S3 y especificar la ubicación del bucket al crear el complemento. Al crear un conector, debe especificar el complemento que desea que MSK Connect utilice para este. La relación entre los complementos y los conectores es la one-to-many siguiente: puede crear uno o más conectores a partir del mismo complemento.

Para obtener información sobre cómo desarrollar el código de un conector, consulte la Guía de desarrollo de conectores en la documentación de Apache Kafka.

Crear un complemento personalizado mediante el AWS Management Console

1. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/.
2. En el panel izquierdo, en MSK Connect, seleccione Complementos personalizados.
3. Seleccione Crear complemento personalizado.
4. Elija Browse S3 (Examinar S3).
5. En la lista de buckets de S3, elija el bucket que contenga el archivo JAR o ZIP del complemento.
6. En la lista de objetos, seleccione la casilla situada a la izquierda del archivo JAR o ZIP del complemento y, a continuación, seleccione Elegir.
7. Seleccione Crear complemento personalizado.

Para usar la API MSK Connect para crear un complemento personalizado, consulte CreateCustomPlugin.

Descripción de los procesos de trabajo de MSK Connect

Un proceso de trabajo es un proceso de máquina virtual Java (JVM) que ejecuta la lógica del conector. Cada proceso de trabajo crea un conjunto de tareas que se ejecutan en subprocesos paralelos y realizan el trabajo de copiar los datos. Las tareas no almacenan el estado y, por lo tanto, se pueden iniciar, detener o reiniciar en cualquier momento para proporcionar una canalización de datos flexible y escalable. Los demás procesos de trabajo detectan automáticamente los cambios en el número de procesos de trabajo, ya sea debido a un problema de escalamiento o a fallos inesperados. Se coordinan para reequilibrar las tareas entre el conjunto de procesos de trabajo restantes. Los procesos de trabajo de Connect utilizan los grupos de consumidores de Apache Kafka para coordinarse y reequilibrarse.

Si los requisitos de capacidad de su conector son variables o difíciles de estimar, puede dejar que MSK Connect escale el número de procesos de trabajo según sea necesario entre el límite inferior y el límite superior que especifique. Como alternativa, puede especificar el número exacto de procesos de trabajo en los que desea ejecutar la lógica del conector. Para obtener más información, consulte the section called “Descripción de la capacidad de los conectores”.

Los procesos de trabajo de MSK Connect consumen direcciones IP

Los procesos de trabajo de MSK Connect consumen direcciones IP en las subredes proporcionadas por el cliente. Cada proceso de trabajo utiliza una dirección IP de una de las subredes

proporcionadas por el cliente. Asegúrese de contar con suficientes direcciones IP disponibles en las subredes proporcionadas a una CreateConnector solicitud para cubrir la capacidad especificada, especialmente al escalar automáticamente los conectores, donde el número de procesos de trabajo puede fluctuar.

Configuración predeterminada del proceso de trabajo

MSK Connect proporciona la siguiente configuración de proceso de trabajo predeterminada:

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

Propiedades de configuración de proceso de trabajo compatibles

MSK Connect proporciona una configuración de proceso de trabajo predeterminada. También tiene la opción de crear una configuración de proceso de trabajo personalizada para utilizarla con sus conectores. La siguiente lista incluye información sobre las propiedades de configuración de proceso de trabajo que Amazon MSK Connect admite o no admite.

- Solo se necesitan las propiedades `key.converter` y `value.converter`.
- MSK Connect admite las siguientes propiedades de configuración de `producer.`.

  ```
  producer.acks
  producer.batch.size
  producer.buffer.memory
  producer.compression.type
  producer.enable.idempotence
  producer.key.serializer
  producer.linger.ms
  producer.max.request.size
  producer.metadata.max.age.ms
  producer.metadata.max.idle.ms
  producer.partitioner.class
  producer.reconnect.backoff.max.ms
  producer.reconnect.backoff.ms
  producer.request.timeout.ms
  producer.retry.backoff.ms
  producer.value.serializer
  ```

- MSK Connect admite las siguientes propiedades de configuración de `consumer.`.

```
consumer.allow.auto.create.topics
consumer.auto.offset.reset
consumer.check.crcs
consumer.fetch.max.bytes
consumer.fetch.max.wait.ms
consumer.fetch.min.bytes
consumer.heartbeat.interval.ms
consumer.key.deserializer
consumer.max.partition.fetch.bytes
consumer.max.poll.interval.ms
consumer.max.poll.records
consumer.metadata.max.age.ms
consumer.partition.assignment.strategy
consumer.reconnect.backoff.max.ms
consumer.reconnect.backoff.ms
consumer.request.timeout.ms
consumer.retry.backoff.ms
consumer.session.timeout.ms
consumer.value.deserializer
```

- Se admiten todas las demás propiedades de configuración que no comiencen por los prefijos `producer.` o `consumer.`, excepto las siguientes propiedades.

```
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
```

Para obtener más información sobre las propiedades de configuración de trabajo y lo que representan, consulte Kafka Connect Configs en la documentación de Apache Kafka.

Creación de una configuración de proceso de trabajo personalizada

En este procedimiento, se describe cómo crear una configuración de proceso de trabajo personalizada utilizando la AWS Management Console.

Creación de una configuración de proceso de trabajo personalizada con AWS Management Console

1. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/.
2. En el panel izquierdo, en MSK Connect, seleccione Configuraciones de proceso de trabajo.
3. Elija Crear configuración de proceso de trabajo.
4. Introduzca un nombre y una descripción opcional y, a continuación, añada las propiedades y los valores en los que desee establecerlos.
5. Elija Crear configuración de proceso de trabajo.

Para usar la API de MSK Connect para crear una configuración del proceso de trabajo, consulte CreateWorkerConfiguration.

Administración de los desplazamientos de los conectores de origen con `offset.storage.topic`

En esta sección se proporciona información que le ayudará a gestionar los desplazamientos de los conectores de origen mediante el tema de almacenamiento de desplazamientos. El tema del

almacenamiento de desplazamientos es un tema interno que Kafka Connect utiliza para almacenar los desplazamientos de configuración de conectores y tareas.

Consideraciones

Tenga en cuenta lo siguiente cuando gestione los desplazamientos del conector de origen.

- Para especificar un tema de almacenamiento de desplazamientos, proporcione el nombre del tema de Kafka en el que se almacenan los desplazamientos de los conectores como valor `offset.storage.topic` en su configuración de proceso de trabajo.
- Tenga cuidado al realizar cambios en la configuración de un conector. El cambio de los valores de configuración puede provocar un comportamiento no deseado del conector si un conector de origen utiliza valores de la configuración para introducir registros de desplazamiento. Recomendamos que consulte la documentación de su complemento para obtener orientación.
- Personalice el número predeterminado de particiones: además de personalizar la configuración de proceso de trabajo añadiendo `offset.storage.topic`, puede personalizar el número de particiones para los temas de compensación y almacenamiento de estado. Las particiones predeterminadas para los temas internos son las siguientes.
  - `config.storage.topic`: 1, no configurable, debe ser un tema de partición única
  - `offset.storage.topic`: 25, configurable proporcionando `offset.storage.partitions`
  - `status.storage.topic`: 5, configurable proporcionando `status.storage.partitions`
- Eliminación manual de temas: Amazon MSK Connect crea nuevos temas internos de Kafka Connect (el nombre del tema comienza por `__amazon_msk_connect`) en cada implementación de conectores. Los temas antiguos que se adjuntan a los conectores eliminados no se eliminan automáticamente, ya que los temas internos, como `offset.storage.topic`, se pueden reutilizar entre los conectores. Sin embargo, puede eliminar manualmente los temas internos no utilizados creados por MSK Connect. Los temas internos se nombran siguiendo el formato `__amazon_msk_connect_<offsets|status|configs>_`*`connector_name_connector_id`*.

  La expresión regular `__amazon_msk_connect_<offsets|status|configs>_`*`connector_name_connector_id`* se puede utilizar para eliminar los temas internos. No debe eliminar un tema interno que utiliza actualmente un conector en ejecución.
- Uso del mismo nombre para los temas internos creados por MSK Connect: si desea reutilizar el tema de almacenamiento de desplazamientos para consumir los desplazamientos de un conector creado anteriormente, debe asignar al nuevo conector el mismo nombre que al conector

anterior. La propiedad `offset.storage.topic` se puede establecer mediante la configuración del proceso de trabajo para asignar el mismo nombre al conector `offset.storage.topic` y reutilizarla entre distintos conectores. Esta configuración se describe en Gestión de desplazamientos de los conectores. MSK Connect no permite que diferentes conectores compartan `config.storage.topic` y `status.storage.topic`. Estos temas se crean cada vez que se crea un conector nuevo en MSKC. Se les asigna un nombre automáticamente según el formato `__amazon_msk_connect_<status|configs>_`*`connector_name_connector_id`* y, por lo tanto, son diferentes en los distintos conectores que cree.

Uso del tema de almacenamiento de desplazamiento predeterminado

De forma predeterminada, Amazon MSK Connect genera un nuevo tema de almacenamiento de desplazamiento en el clúster de Kafka para cada conector que cree. MSK crea el nombre del tema por defecto utilizando partes del ARN del conector. Por ejemplo, `__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-4abc-8be8-c657f7e4ff32-2`.

Uso del tema de almacenamiento de desplazamiento personalizado

Para proporcionar continuidad de desplazamiento entre los conectores de origen, puede utilizar un tema de almacenamiento de desplazamiento de su elección en lugar del tema predeterminado. Especificar un tema de almacenamiento de desplazamiento le ayuda a realizar tareas como crear un conector de origen que reanude la lectura desde el último desplazamiento de un conector anterior.

Para especificar un tema de almacenamiento de desplazamiento, debe proporcionar un valor para la propiedad `offset.storage.topic` en su configuración de proceso de trabajo antes de crear un conector. Si desea reutilizar el tema de almacenamiento de desplazamientos para consumir los desplazamientos de un conector creado anteriormente, debe asignar al nuevo conector el mismo nombre que al conector anterior. Si crea un tema de almacenamiento de desplazamiento personalizado, debe definir `cleanup.policy` como `compact` en la configuración del tema.

> **Note**
>
> Si especifica un tema de almacenamiento de desplazamiento al crear un conector de recepción, MSK Connect crea el tema si aún no existe. Sin embargo, el tema no se utilizará para almacenar los desplazamientos de los conectores.
> En cambio, los desplazamientos de los conectores de recepción se gestionan mediante el protocolo de grupos de consumidores de Kafka. Cada conector de recepción crea un grupo

denominado `connect-{CONNECTOR_NAME}`. Mientras exista el grupo de consumidores, cualquier conector de recepción sucesivo que se cree con el mismo valor `CONNECTOR_NAME` se mantendrá desde el último desplazamiento asignado.

Example : especificación de un tema de almacenamiento de desplazamiento para recrear un conector de origen con una configuración actualizada

Supongamos que tiene un conector de captura de datos modificados (CDC) y desea modificar la configuración del conector sin perder su lugar en la transmisión de CDC. No puede actualizar la configuración de conector existente, pero puede eliminar el conector y crear uno nuevo con el mismo nombre. Para indicar al nuevo conector por dónde empezar a leer en la transmisión de CDC, puede especificar el tema de almacenamiento de desplazamiento del conector anterior en su configuración de proceso de trabajo. En los siguientes pasos se muestra cómo realizar esta tarea.

1. En el equipo cliente, ejecute el siguiente comando para buscar el nombre del tema de almacenamiento de desplazamiento del conector. Sustituya *<bootstrapBrokerString>* por la cadena de agente de arranque de su clúster. Para ver instrucciones sobre cómo obtener la cadena de su agente de arranque, consulte Obtención de agentes de arranque para un clúster de Amazon MSK.

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

   El siguiente resultado muestra una lista de todos los temas del clúster, incluidos los temas de conectores internos predeterminados. En este ejemplo, el conector CDC existente utiliza el tema de almacenamiento de desplazamiento predeterminado creado por MSK Connect. Por eso el tema del almacenamiento de desplazamiento se denomina `__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-4abc-8be8-c657f7e4ff32-2`.

   ```
   __consumer_offsets
   __amazon_msk_canary
   __amazon_msk_connect_configs_my-mskc-connector_12345678-09e7-4abc-8be8-
   c657f7e4ff32-2
   __amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-4abc-8be8-
   c657f7e4ff32-2
   __amazon_msk_connect_status_my-mskc-connector_12345678-09e7-4abc-8be8-
   c657f7e4ff32-2
   ```

```
my-msk-topic-1
my-msk-topic-2
```

2. Abra la consola de Amazon MSK en https://console.aws.amazon.com/msk/.
3. Elija el conector de la lista Conectores. Copie y guarde el contenido del campo de Configuración del conector para poder modificarlo y usarlo para crear el nuevo conector.
4. Para eliminar el conector, elija Eliminar. A continuación, ingrese el nombre del conector en el campo de entrada de texto para confirmar la eliminación.
5. Cree una configuración de proceso de trabajo personalizada con valores que se adapten a su caso de uso. Para obtener instrucciones, consulte Creación de una configuración de proceso de trabajo personalizada.

   En su configuración de proceso de trabajo, debe especificar el nombre del tema de almacenamiento de desplazamiento que ha recuperado anteriormente como valor para `offset.storage.topic` como en la siguiente configuración.

   ```
   config.providers.secretManager.param.aws.region=eu-west-3
   key.converter=<org.apache.kafka.connect.storage.StringConverter>
   value.converter=<org.apache.kafka.connect.storage.StringConverter>
   config.providers.secretManager.class=com.github.jcustenborder.kafka.config.aws.SecretsManag
   config.providers=secretManager
   offset.storage.topic=__amazon_msk_connect_offsets_my-mskc-
   connector_12345678-09e7-4abc-8be8-c657f7e4ff32-2
   ```

6. **Important**

   Debe asignar al conector nuevo el mismo nombre que al conector anterior.

   Cree un conector nuevo con la configuración de proceso de trabajo que configuró en el paso anterior. Para obtener instrucciones, consulte Crear un conector.

Tutorial: Externalización de la información confidencial con proveedores de configuración

En este ejemplo, se muestra cómo externalizar la información confidencial de Amazon MSK Connect mediante un proveedor de configuración de código abierto. Un proveedor de configuración le permite

especificar variables en lugar de texto sin formato en una configuración de conector o de trabajo, y los procesos de trabajo que se ejecutan en su conector resuelven estas variables en tiempo de ejecución. Esto evita que las credenciales y otros secretos se almacenen en texto sin formato. El proveedor de configuración del ejemplo admite la recuperación de los parámetros de configuración de AWS Secrets Manager, Amazon S3 y Systems Manager (SSM). En el paso 2, puede ver cómo configurar el almacenamiento y la recuperación de información confidencial para el servicio que desee configurar.

Consideraciones

Tenga en cuenta lo siguiente al utilizar el proveedor de configuración de MSK con Amazon MSK Connect:

- Al utilizar los proveedores de configuración, asigne los permisos adecuados al rol de ejecución del servicio de IAM.
- Defina los proveedores de configuración en las configuraciones de proceso de trabajo y su implementación en la configuración del conector.
- Los valores de configuración confidenciales pueden aparecer en los registros de los conectores si un complemento no los define como secretos. Kafka Connect trata los valores de configuración indefinidos de la misma manera que cualquier otro valor de texto sin formato. Para obtener más información, consulte Impedir que los secretos aparezcan en los registros de los conectores.
- De forma predeterminada, MSK Connect reinicia con frecuencia un conector cuando este utiliza un proveedor de configuración. Para desactivar este comportamiento de reinicio, puede establecer el valor de `config.action.reload` en `none` en la configuración del conector.

Creación de un complemento personalizado y carga a S3

Para crear un complemento personalizado, cree un archivo zip que contenga el conector y ejecute msk-config-provider los siguientes comandos en su máquina local.

Para crear un complemento personalizado utilizando una ventana de terminal y Debezium como conector

Utilice la AWS CLI para ejecutar comandos como superusuario con credenciales que le permitan acceder a su bucket de AWS S3. Para obtener información sobre la instalación y configuración de la AWS CLI, consulte Introducción a la AWS CLI en la Guía del AWS Command Line Interface usuario.

Para obtener información sobre el uso de la AWS CLI con Amazon S3, consulte Uso de Amazon S3 con la AWS CLI en la Guía del AWS Command Line Interface usuario.

1. En una ventana de terminal, cree una carpeta denominada `custom-plugin` en su espacio de trabajo mediante el siguiente comando.

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

2. Descargue la última versión estable del complemento MySQL Connector desde el sitio de Debezium mediante el siguiente comando.

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

   Extraiga el archivo gzip descargado de la carpeta `custom-plugin` con el siguiente comando.

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

3. Descargue el archivo zip del proveedor de configuración de MSK con el siguiente comando.

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

   Extraiga el archivo zip descargado de la carpeta `custom-plugin` con el siguiente comando.

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

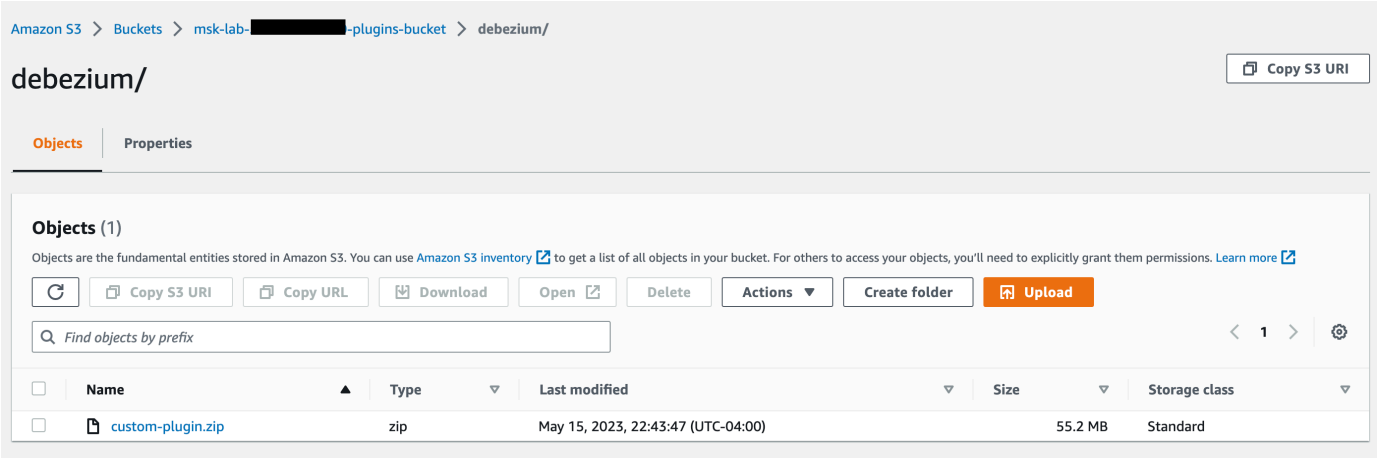
4. Comprima el contenido del proveedor de configuración de MSK del paso anterior y el conector personalizado en un único archivo denominado `custom-plugin.zip`.

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

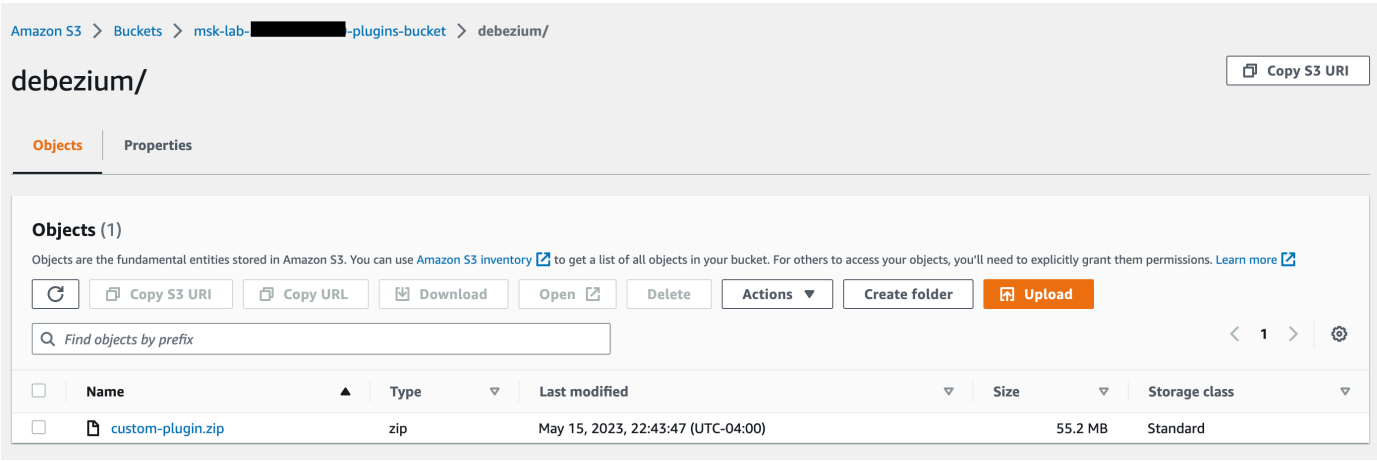
5. Cargue el archivo en S3 para consultarlo más adelante.

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

6. En la consola de Amazon MSK, en la sección MSK Connect, elija Custom Plugin, elija Create custom plugin y explore el bucket s3: < ***S3_URI_BUCKET_LOCATION*** > S3 para seleccionar el archivo ZIP del plugin personalizado que acaba de cargar.



7. Introduzca **`debezium-custom-plugin`** para el nombre del complemento. De manera opcional, ingrese una descripción y elija Crear complemento personalizado.



Configuración de los parámetros y permisos para los distintos proveedores

Puede configurar los valores de los parámetros en estos tres servicios:

- Secrets Manager
- Almacén de parámetros de Systems Manager
- S3 - Simple Storage Service

Seleccione una de las pestañas siguientes para obtener instrucciones sobre cómo configurar los parámetros y los permisos pertinentes para ese servicio.

Configure in Secrets Manager

Configuración de valores de parámetros en Secrets Manager

1. Abra la consola de Secrets Manager.
2. Cree un nuevo secreto para almacenar sus credenciales o secretos. Para obtener instrucciones, consulte Crear un AWS Secrets Manager secreto en la Guía del AWS Secrets Manager usuario.
3. Copie el ARN de su secreto.
4. Añada los permisos de Secrets Manager de la siguiente política de ejemplo al rol de ejecución del servicio. *<arn:aws:secretsmanager:us-east-1:123456789000:secret:MySecret-1234>*Sustitúyalo por el ARN de su secreto.
5. Agregue la configuración de proceso de trabajo y las instrucciones del conector.

```
{
        "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. Para utilizar el proveedor de configuración Secrets Manager, copie las siguientes líneas de código en el cuadro de texto de configuración del proceso de trabajo del paso 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. Para el proveedor de configuración Secrets Manager, copie las siguientes líneas de código en la configuración del conector del paso 4.

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

También puede utilizar el paso anterior con más proveedores de configuración.

Configure in Systems Manager Parameter Store

Configuración de valores de parámetros en el almacén de parámetros de Systems Manager

1. Abra la consola de Systems Manager.
2. En el panel de navegación, elija Parameter Store (Almacén de parámetros).
3. Cree un parámetro nuevo para guardarlo en Systems Manager. Para obtener instrucciones, consulte Creación de un parámetro de Systems Manager (consola) en la Guía del AWS Systems Manager usuario.
4. Copie el ARN de su parámetro.
5. Añada los permisos de Systems Manager de la siguiente política de ejemplo al rol de ejecución del servicio. *<arn:aws:ssm:us-east-1:123456789000:parameter/MyParameterName>*Sustitúyalo por el ARN de su parámetro.

```
{
        "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. Para utilizar el proveedor de configuración del almacén de parámetros, copie las siguientes líneas de código en el cuadro de texto de configuración del proceso de trabajo del paso 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. Para el proveedor de configuración del almacén de parámetros, copie las siguientes líneas de código en la configuración del conector del paso 5.

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

También puede agrupar los dos pasos anteriores con más proveedores de configuración.

Configure in Amazon S3

Configuración de objetos o archivos en Amazon S3

1. Abra la consola de Amazon S3.

2. Cargue su objeto en un bucket en S3. Para obtener instrucciones, consulte Carga de objetos.
3. Copie el ARN de su objeto.
4. Añada los permisos de lectura de objetos de Amazon S3 de la siguiente política de ejemplo al rol de ejecución del servicio. *<arn:aws:s3:::MY_S3_BUCKET/path/to/custom-plugin.zip>*Sustitúyalo por el ARN de su objeto.

```
{
    "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. Para utilizar el proveedor de configuración de Amazon S3, copie las siguientes líneas de código en el cuadro de texto de configuración del proceso de trabajo del paso 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. Para el proveedor de configuración de Amazon S3, copie las siguientes líneas de código en la configuración del conector del paso 4.

```
#Example implementation for S3 object

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

También puede agrupar los dos pasos anteriores con más proveedores de configuración.

Creación de una configuración de proceso de trabajo personalizada con información sobre su proveedor de configuración

1. Seleccione Configuraciones de proceso de trabajo en la sección Amazon MSK Connect.
2. Seleccione Crear configuración de proceso de trabajo.
3. Introduzca `SourceDebeziumCustomConfig` en el cuadro de texto Nombre de configuración de proceso de trabajo. La descripción es opcional.
4. Copie el código de configuración correspondiente en función de los proveedores que desee y péguelo en el cuadro de texto Configuración del proceso de trabajo.
5. Este es un ejemplo de la configuración de proceso de trabajo para los tres proveedores:

```
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. Haga clic en Crear configuración del proceso de trabajo.

Creación del conector

1. Cree un conector nuevo siguiendo las instrucciones de Crear un conector nuevo.
2. Elija el archivo `custom-plugin.zip` que cargó en su bucket de S3 en ??? como origen del complemento personalizado.
3. Copie el código de configuración correspondiente en función de los proveedores que desee y péguelo en el cuadro de texto Configuración del conector.
4. Este es un ejemplo de la configuración del conector para los tres proveedores:

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

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

#Example implementation for Amazon S3 file/object
database.ssl.truststore.location = ${s3import:us-west-2:my_cert_bucket/path/to/
trustore_unique_filename.jks}
```

5. Seleccione Usar una configuración personalizada y elija SourceDebeziumCustomConfiguna opción en el menú desplegable Configuración de trabajo.
6. Siga los pasos restantes de las instrucciones de Crear un conector.

Políticas y roles de IAM para MSK Connect

Esta sección le ayuda a configurar las políticas y funciones de IAM adecuadas para implementar y gestionar Amazon MSK Connect de forma segura en su AWS entorno. En las siguientes secciones, se explica el rol de ejecución de servicios que debe utilizarse con MSK Connect, incluida la política de confianza requerida y los permisos adicionales necesarios para conectarse a un clúster de MSK con autenticación de IAM. En esta página también se proporcionan ejemplos de políticas de IAM integrales para garantizar el acceso total a la funcionalidad de MSK Connect, así como detalles sobre las políticas AWS administradas de disponibles para el servicio.

Temas

- Descripción del rol de ejecución de servicios
- Ejemplo de política de IAM para MSK Connect

- Prevención de los problemas del suplente confuso entre servicios
- AWS Políticas administradas por para MSK Connect
- Uso de roles vinculados a servicios para MSK Connect

Descripción del rol de ejecución de servicios

> **Note**
>
> Amazon MSK Connect no admite el uso del rol vinculado al servicio como rol de ejecución del servicio. Debe crear un rol de ejecución del servicio independiente. Para obtener instrucciones sobre cómo crear un rol de IAM personalizado, consulte Crear un rol para delegar permisos a un AWS servicio en la Guía del usuario de IAM.

Al crear un conector con MSK Connect, debe especificar un rol de AWS Identity and Access Management (IAM) para usarlo con él. Su rol de ejecución del servicio debe tener la siguiente política de confianza para que MSK Connect pueda asumirla. Para obtener información sobre las claves de contexto de condición en esta política, consulte the section called “Prevención de los problemas del suplente confuso entre servicios”.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "kafkaconnect.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:kafkaconnect:us-east-1:123456789012:connector/
myConnector/abc12345-abcd-4444-a8b9-123456f513ed-2"
        }
      }
    }
```

```
  ]
}
```

Si el clúster de Amazon MSK que desea utilizar con su conector es un clúster que utiliza la autenticación de IAM, debe añadir la siguiente política de permisos al rol de ejecución del servicio del conector. Para obtener información acerca de cómo encontrar el UID de clúster y cómo crear el tema ARNs, consultethe section called “Recursos de la política de autorización”.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:DescribeCluster"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:000000000001:cluster/
testClusterName/300d0000-0000-0005-000f-00000000000b-1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:ReadData",
                "kafka-cluster:DescribeTopic"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:123456789012:topic/
myCluster/300a0000-0000-0003-000a-00000000000b-6/__amazon_msk_connect_read"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:WriteData",
                "kafka-cluster:DescribeTopic"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:123456789012:topic/
testCluster/300f0000-0000-0008-000d-00000000000m-7/__amazon_msk_connect_write"
            ]
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:CreateTopic",
                "kafka-cluster:WriteData",
                "kafka-cluster:ReadData",
                "kafka-cluster:DescribeTopic"
            ],
            "Resource": [
                "arn:aws:kafka:us-
east-1:123456789012:topic/testCluster/300f0000-0000-0008-000d-00000000000m-7/
__amazon_msk_connect_*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeGroup"
            ],
            "Resource": [
                "arn:aws:kafka:us-
east-1:123456789012:group/testCluster/300d0000-0000-0005-000f-00000000000b-1/
__amazon_msk_connect_*",
                "arn:aws:kafka:us-
east-1:123456789012:group/testCluster/300d0000-0000-0005-000f-00000000000b-1/connect-*"
            ]
        }
    ]
}
```

Según el tipo de conector, es posible que también tenga que adjuntar al rol de ejecución del servicio una política de permisos que le permita acceder a AWS los recursos de. Por ejemplo, si el conector necesita enviar datos a un bucket de S3, el rol de ejecución del servicio debe tener una política de permisos que conceda permiso para escribir en ese bucket. Para realizar pruebas, puede usar una de las políticas de IAM prediseñadas que otorgan acceso total, como `arn:aws:iam::aws:policy/AmazonS3FullAccess`. Sin embargo, por motivos de seguridad, le recomendamos que utilice la política más restrictiva que permita al conector leer desde el AWS origen de o escribir en el receptor AWS de.

Ejemplo de política de IAM para MSK Connect

Para dar a un usuario que no sea administrador acceso completo a todas las funciones de MSK Connect, adjunte una política como la siguiente al rol de IAM del usuario.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MSKConnectFullAccess",
      "Effect": "Allow",
      "Action": [
        "kafkaconnect:CreateConnector",
        "kafkaconnect:DeleteConnector",
        "kafkaconnect:DescribeConnector",
        "kafkaconnect:GetConnector",
        "kafkaconnect:ListConnectors",
        "kafkaconnect:UpdateConnector",
        "kafkaconnect:CreateCustomPlugin",
        "kafkaconnect:DeleteCustomPlugin",
        "kafkaconnect:DescribeCustomPlugin",
        "kafkaconnect:GetCustomPlugin",
        "kafkaconnect:ListCustomPlugins",
        "kafkaconnect:CreateWorkerConfiguration",
        "kafkaconnect:DeleteWorkerConfiguration",
        "kafkaconnect:DescribeWorkerConfiguration",
        "kafkaconnect:GetWorkerConfiguration",
        "kafkaconnect:ListWorkerConfigurations"
      ],
      "Resource": "*"
    },
    {
      "Sid": "IAMPassRole",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/MSKConnectServiceRole",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "kafkaconnect.amazonaws.com"
        }
      }
    },
    {
```

```
      "Sid": "EC2NetworkAccess",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface",
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups"
      ],
      "Resource": "*"
    },
    {
      "Sid": "MSKClusterAccess",
      "Effect": "Allow",
      "Action": [
        "kafka:DescribeCluster",
        "kafka:DescribeClusterV2",
        "kafka:GetBootstrapBrokers"
      ],
      "Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/myCluster/"
    },
    {
      "Sid": "MSKLogGroupAccess",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams",
        "logs:DescribeLogGroups"
      ],
      "Resource": [
        "arn:aws:logs:us-east-1:123456789012:log-group:/aws/msk-connect/*"
      ]
    },
    {
      "Sid": "S3PluginAccess",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
      ],
```

```
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1-custom-plugins",
        "arn:aws:s3:::amzn-s3-demo-bucket1-custom-plugins/*"
      ]
    }
  ]
}
```

Prevención de los problemas del suplente confuso entre servicios

El problema de la sustitución confusa es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación de identidad entre servicios puede provocar un confuso problema de diputado. La suplantación entre servicios puedes producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puedes manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que lo ayudan a proteger sus datos para todos los servicios con entidades principales de servicio a las que se les ha dado acceso a los recursos de su cuenta.

Se recomienda utilizar las claves de contexto de condición global `aws:SourceArn` y `aws:SourceAccount` en las políticas de recursos para limitar los permisos que MSK Connect concede a otro servicio para el recurso. Si el valor de `aws:SourceArn` no contiene el ID de cuenta (por ejemplo, un ARN de bucket de Amazon S3 no contiene el ID de cuenta), debe utilizar ambas claves de contexto de condición global para limitar los permisos. Si utiliza claves de contexto de condición global y el valor de `aws:SourceArn` contiene el ID de cuenta, el valor de `aws:SourceAccount` y la cuenta en el valor de `aws:SourceArn` deben utilizar el mismo ID de cuenta cuando se utiliza en la misma instrucción de política. Utiliza `aws:SourceArn` si desea que solo se asocie un recurso al acceso entre servicios. Utiliza `aws:SourceAccount` si quiere permitir que cualquier recurso de esa cuenta se asocie al uso entre servicios.

En el caso de MSK Connect, el valor de `aws:SourceArn` debe ser un conector de MSK.

La forma más eficaz de protegerse contra el problema de la sustitución confusa es utilizar la clave de contexto de condición global de `aws:SourceArn` con el ARN completo del recurso. Si no conoce el ARN completo del recurso o si especifica varios recursos, utiliza la clave de condición de contexto global `aws:SourceArn` con comodines (*) para las partes desconocidas del ARN. Por ejemplo, *arn:aws:kafkaconnect:us-east-1:123456789012:connector/** representa todos los

conectores que pertenecen a la cuenta con el identificador 123456789012 en la región EE.UU. Este (Norte de Virginia).

El siguiente ejemplo muestra cómo se pueden utilizar las claves contextuales de condición global `aws:SourceArn` y `aws:SourceAccount` en MSK Connect para evitar el problema del adjunto confundido. Reemplace el *`Account-ID`* y el *`MSK-Connector-ARN`* con su información.

```
{
  "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 Políticas administradas por para MSK Connect

Una política AWS administrada por es una política independiente creada y administrada por AWS. AWS Las políticas administradas de se han diseñado para ofrecer permisos para muchos casos de uso habituales, por lo que puede empezar a asignar permisos a los usuarios, grupos y roles.

Considere que es posible que las políticas AWS administradas de no concedan permisos de privilegio mínimo para los casos de uso concretos, ya que están disponibles para que las utilicen todos los AWS clientes de. Se recomienda definir políticas administradas por el cliente específicas para sus casos de uso a fin de reducir aún más los permisos.

No puede cambiar los permisos definidos en las políticas AWS administradas de. Si AWS actualiza los permisos definidos en una política AWS administrada de, la actualización afecta a todas las

identidades principales (usuarios, grupos y roles) a las que está adjunta la política. AWS es más probable que actualice una política AWS gestionada por cuando Servicio de AWS se lanza una nueva o cuando estén disponibles operaciones de API nuevas para los servicios existentes.

Para obtener más información, consulte Políticas administradas de AWS en la Guía del usuario de IAM.

AWS política gestionada: Amazon MSKConnect ReadOnlyAccess

Esta política otorga al usuario los permisos necesarios para enumerar y describir los recursos de MSK Connect.

Puede adjuntar la política `AmazonMSKConnectReadOnlyAccess` a las identidades de IAM.

```
{
    "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 política gestionada: KafkaConnectServiceRolePolicy

Esta política otorga al servicio MSK Connect los permisos necesarios para crear y administrar las interfaces de red que tienen la etiqueta `AmazonMSKConnectManaged:true`. Estas interfaces de red proporcionan acceso de red de MSK Connect a los recursos de su Amazon VPC, como un clúster de Apache Kafka o un origen o un receptor.

No puede conectarse KafkaConnectServiceRolePolicy a sus entidades de IAM. Esta política está adjunta a un rol vinculado a servicios que permite a MSK Connect realizar acciones en su nombre.

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

Actualizaciones de MSK Connect para las políticas AWS administradas por

Es posible consultar los detalles sobre las actualizaciones de las políticas AWS administradas por para MSK Connect debido a que este servicio comenzó a hacer un seguimiento de estos cambios.

Cambio	Descripción	Fecha
MSK Connect actualizó la política de solo lectura	MSK Connect actualizó la MSKConnect ReadOnlyA ccess política de Amazon para eliminar las restricciones a las operaciones de publicación de anuncios.	13 de octubre de 2021
MSK Connect comenzó a realizar un seguimiento de los cambios	MSK Connect comenzó a hacer un seguimiento de los cambios de las políticas AWS administradas por.	14 de septiembre de 2021

Uso de roles vinculados a servicios para MSK Connect

Amazon MSK Connect utiliza roles vinculados a servicios de AWS Identity and Access Management (IAM). Un rol vinculado a servicios es un tipo único de rol de IAM que está vinculado directamente a MSK Connect. Los roles vinculados a servicios están predefinidos por MSK Connect e incluyen todos los permisos que el servicio requiere para llamar a AWS otros servicios de en su nombre.

Un rol vinculado a un servicio simplifica la configuración de MSK Connect porque ya no tendrá que agregar manualmente los permisos necesarios. MSK Connect define los permisos de sus roles vinculados a servicios y, a menos que esté definido de otra manera, solo MSK Connect puede asumir sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda asociar a ninguna otra entidad de IAM.

Para obtener información acerca de otros servicios que admiten roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Yes (Sí) en la columna Service Linked Role (Rol vinculado a servicios). Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado a servicios en cuestión.

Permisos de roles vinculados a servicios de MSK Connect

MSK Connect utiliza el rol vinculado al servicio denominado: AWSServiceRoleForKafkaConnectpermite que Amazon MSK Connect acceda a los recursos de Amazon en su nombre.

El rol AWSService RoleForKafkaConnect vinculado al servicio confía en el `kafkaconnect.amazonaws.com` servicio para asumir el rol.

Para obtener información sobre la política de permisos que utiliza el rol, consulte the section called "KafkaConnectServiceRolePolicy".

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o rol) crear, editar o eliminar un rol vinculado a servicios. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Creación de un rol vinculado al servicio para MSK Connect

No necesita crear manualmente un rol vinculado a servicios. Al crear un conector en la AWS Management Console, la, o la AWS API de AWS CLI, MSK Connect crea el rol vinculado al servicio en su nombre.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Al crear un conector, MSK Connect se encarga de crear de nuevo el rol vinculado al servicio por usted.

Modificación de un rol vinculado al servicio para MSK Connect

MSK Connect no permite editar el rol vinculado al AWSService RoleForKafkaConnect servicio. Después de crear un rol vinculado a un servicio, no puede cambiarle el nombre, ya que varias entidades pueden hacer referencia a él. Sin embargo, puede editar la descripción del rol mediante IAM. Para obtener más información, consulte Edición de un rol vinculado a servicios en la Guía del usuario de IAM.

Eliminación de un rol vinculado a un servicio en MSK Connect

Puede utilizar la consola de IAM, la AWS CLI o la AWS API de para eliminar manualmente el rol vinculado al servicio. Para ello, primero elimine manualmente todos los conectores de MSK Connect y, a continuación, elimine manualmente el rol. Para obtener más información, consulte Eliminación de un rol vinculado a servicios en la Guía del usuario de IAM.

Regiones admitidas para los roles vinculados al servicio de MSK Connect

MSK Connect admite el uso de roles vinculados al servicio en todas las regiones en las que el servicio esté disponible. Para obtener más información, consulte Regiones y puntos de conexión de AWS.

Habilitación del acceso a Internet para Amazon MSK Connect

Si su conector para Amazon MSK Connect necesita acceso a Internet, le recomendamos que utilice la siguiente configuración Amazon Virtual Private Cloud (VPC) para habilitar ese acceso.

- Configure el conector con subredes privadas.
- Cree una puerta de enlace NAT pública o una instancia de NAT para su VPC en una subred pública. Para obtener más información, consulte la página Conectar subredes a Internet u otros dispositivos VPCs mediante NAT en la Guía del Amazon Virtual Private Cloudusuario.
- Permita el tráfico saliente de sus subredes privadas a su instancia o puerta de enlace NAT.

Configuración de una puerta de enlace NAT para Amazon MSK Connect

Los pasos siguientes muestran cómo configurar una puerta de enlace NAT para habilitar el acceso a Internet de un conector. Debe completar estos pasos antes de crear un conector en una subred privada.

Complete los requisitos previos para configurar una puerta de enlace NAT

Asegúrese de tener los siguientes elementos.

- El ID de la Amazon Virtual Private Cloud (VPC) asociada al clúster. Por ejemplo, vpc-123456ab.
- Las subredes privadas IDs de su VPC. Por ejemplo, subnet-a1b2c3de, subnet-f4g5h6ij, etc. Debe configurar el conector con subredes privadas.

Pasos para habilitar el acceso a Internet para su conector

Habilitación del acceso a Internet para su conector

1. Abra la Amazon Virtual Private Cloud consola en. https://console.aws.amazon.com/vpc/
2. Cree una subred pública para su puerta de enlace NAT con un nombre descriptivo y anote el ID de subred. Para obtener instrucciones detalladas, consulte Crear una subred en la VPC.

3. Cree una puerta de enlace de Internet para que su VPC pueda comunicarse con Internet y anote el ID de la puerta de enlace. Adjunte una puerta de enlace de Internet a su VPC. Para obtener más instrucciones, consulte Crear y adjuntar una puerta de enlace de Internet.
4. Aprovisione una puerta de enlace NAT pública para que los hosts de sus subredes privadas puedan acceder a su subred pública. Cuando cree la puerta de enlace NAT, seleccione la subred pública que creó anteriormente. Para obtener instrucciones, consulte Creación de una puerta de enlace NAT.
5. Configure sus tablas de enrutamiento. Debe tener dos tablas de enrutamiento en total para completar esta configuración. Ya debería tener una tabla de enrutamiento principal que se haya creado automáticamente al mismo tiempo que su VPC. En este paso, crea una tabla de enrutamiento adicional para su subred pública.

   a. Utilice la siguiente configuración para modificar la tabla de enrutamiento principal de la VPC de modo que las subredes privadas dirijan el tráfico a la puerta de enlace NAT. Para obtener instrucciones, consulte Trabajar con tablas de enrutamiento en la Guía del usuario de Amazon Virtual Private Cloud.

      Tabla de enrutamiento Private MSKC

Propiedad	Valor
Etiqueta de nombre	Le recomendamos que asigne a esta tabla de enrutamiento una etiqueta descriptiva con su nombre para ayudarlo a identific arla. Por ejemplo, Private MSKC.
Subredes asociadas	Sus subredes privadas
Una ruta para habilitar el acceso a Internet para MSK Connect	• Destino: 0.0.0.0/0 • Destino: su ID de puerta de enlace NAT. Por ejemplo, nat-12a345bc6789efg1h.
Una ruta local para el tráfico interno	• Destino: 10.0.0.0/16. Este valor puede variar en función del bloque CIDR de la VPC. • Objetivo: local

b. Siga las instrucciones en Crear de una tabla de enrutamiento personalizada para crear una para la subred pública. Al crear la tabla, introduzca un nombre descriptivo en el campo Etiqueta de nombre para ayudarle a identificar la subred a la que está asociada la tabla. Por ejemplo, Public MSKC.

c. Configure su tabla de enrutamiento Public MSKC con los siguientes ajustes.

Propiedad	Valor
Etiqueta de nombre	Public MSKC o un nombre descriptivo diferente que elija
Subredes asociadas	Su subred pública con puerta de enlace NAT
Una ruta para habilitar el acceso a Internet para MSK Connect	• Destino: 0.0.0.0/0 • Destino: su ID de puerta de enlace de Internet. Por ejemplo, igw-1a234bc5.
Una ruta local para el tráfico interno	• Destino: 10.0.0.0/16. Este valor puede variar en función del bloque CIDR de la VPC. • Objetivo: local

Descripción de los nombres de host de DNS privados

Con la compatibilidad con nombres de host DNS privados en MSK Connect, puede configurar conectores para que hagan referencia a nombres de dominio públicos o privados. El soporte depende de los servidores DNS especificados en el conjunto de opciones de DHCP de la VPC.

Un conjunto de opciones de DHCP es un grupo de configuraciones de red que las EC2 instancias en la VPC utilizan para comunicarse a través de la red VPC. Cada VPC dispone de un conjunto de opciones de DHCP predeterminado, pero puede crear un conjunto de opciones de DHCP personalizado si, por ejemplo, desea que las instancias de la VPC utilicen un servidor DNS diferente para la resolución de nombres de dominio en lugar del servidor DNS de Amazon. Consulte Conjuntos de opciones de DHCP en Amazon VPC.

Antes de incluir la funcionalidad o característica de resolución de DNS privado en MSK Connect, los conectores utilizaban los solucionadores de DNS de VPC del servicio para las consultas de DNS de un conector de cliente. Los conectores no utilizaban los servidores DNS definidos en los conjuntos de opciones de DHCP de la VPC del cliente para la resolución de DNS.

Los conectores solo podían hacer referencia a los nombres de host en las configuraciones de los conectores del cliente o en los complementos que se pudieran resolver públicamente. No podían resolver los nombres de host privados definidos en una zona alojada de forma privada ni utilizar servidores DNS en la red de otro cliente.

Sin el DNS privado, los clientes que optaran por hacer que sus bases de datos, almacenes de datos y sistemas, como el Secrets Manager de su propia VPC, fueran inaccesibles a Internet, no podían trabajar con los conectores de MSK. Los clientes suelen utilizar nombres de servidor DNS privados para cumplir con las normas de seguridad corporativas.

Configuración de un conjunto de opciones de DHCP de la VPC para su conector

Los conectores utilizan automáticamente los servidores DNS definidos en su conjunto de opciones de DHCP de VPC al crear el conector. Antes de crear un conector, asegúrese de configurar el conjunto de opciones de DHCP de VPC para los requisitos de resolución de nombres de host DNS del conector.

Los conectores que creó antes de que la característica de nombre de host de DNS privado estuviera disponible en MSK Connect siguen utilizando la configuración de resolución de DNS anterior sin necesidad de modificarlos.

Si solo necesita una resolución de nombres de host DNS de resolución pública en su conector, para facilitar la configuración, le recomendamos que utilice la VPC predeterminada de su cuenta al crear el conector. Consulte Servidor DNS de Amazon en la Guía del usuario de Amazon VPC para obtener más información sobre el servidor DNS proporcionado por Amazon o Amazon Route 53 Resolver.

Si necesita resolver los nombres de host DNS privados, asegúrese de que la VPC que se transfiere durante la creación del conector tenga el conjunto de opciones de DHCP configurado correctamente. Para obtener más información, consulte Trabajar con los conjuntos de opciones de DHCP en la Guía del usuario de Amazon VPC.

Al configurar un conjunto de opciones de DHCP para la resolución de nombres de host de DNS privados, asegúrese de que el conector pueda acceder a los servidores DNS personalizados que

configure en el conjunto de opciones de DHCP. De lo contrario, se producirá un error al crear el conector.

Tras personalizar el conjunto de opciones de DHCP de la VPC, los conectores que se creen posteriormente en esa VPC utilizan los servidores DNS que especificó en el conjunto de opciones. Si cambia el conjunto de opciones después de crear un conector, el conector adopta la configuración del nuevo conjunto de opciones en un par de minutos.

Configuración de atributos de DNS para su VPC

Asegúrese de tener los atributos de DNS de la VPC correctamente configurados, tal y como se describe en la sección Atributos de DNS de la VPC y los nombres de host de DNS de la Guía del usuario de Amazon VPC.

Consulte Resolución de consultas de DNS entre VPCs y la red en la Guía para desarrolladores de Amazon Route 53 para obtener información sobre el uso de puntos de conexión de resolución entrantes y salientes para conectar otras redes a su VPC y que funcionen con su conector.

Solución de errores en la creación de conectores

Esta sección describe los posibles errores en la creación de conectores relacionados con la resolución de DNS y sugerencias de acciones para resolver los problemas.

Failure	Acción sugerida
La creación del conector falla si se produce un error en una consulta de resolución de DNS o si no se puede acceder a los servidores DNS desde el conector.	Si ha configurado estos CloudWatch registros para su conector, puede ver errores en la creación de conectores debido a consultas de resolución de DNS fallidas en sus registros. Compruebe las configuraciones del servidor DNS y asegúrese de que la red esté conectada a los servidores DNS desde el conector.
Si cambia la configuración de los servidores DNS en el conjunto de opciones de DHCP de la VPC mientras se ejecuta un conector, las consultas de resolución de DNS del conector pueden fallar. Si se produce un error en la	Si ha configurado estos CloudWatch registros para su conector, puede ver errores en la creación de conectores debido a consultas de resolución de DNS fallidas en sus registros.

Failure	Acción sugerida
resolución de DNS, algunas de las tareas del conector pueden entrar en un estado fallido.	Las tareas fallidas deberían reiniciarse automáticamente para que el conector vuelva a funcionar. Si eso no ocurre, puede ponerse en contacto con el servicio de asistencia para reiniciar las tareas fallidas en el conector correspondiente o puede volver a crear el conector.

Seguridad de MSK Connect

Puede utilizar un punto de conexión de VPC de interfaz, con tecnología de AWS PrivateLink, para evitar que el tráfico entre su Amazon VPC y Amazon MSK Connect APIs compatible salga de la red de Amazon. Los puntos de conexión de VPC de interfaz no requieren una puerta de enlace de Internet, un dispositivo NAT, una conexión de VPN ni una conexión de. AWS Direct Connect Para obtener más información, consulte Utilice Amazon MSK APIs con puntos de enlace de VPC de interfaz.

Registro de MSK Connect

MSK Connect puede escribir eventos de registro que puede usar para depurar el conector. Al crear un conector, puede especificar ninguno o varios de los siguientes destinos de registro:

- Amazon CloudWatch Logs: usted especifica el grupo de registros al que quiere que MSK Connect envíe los eventos de registro de su conector. Para obtener información sobre cómo crear un grupo de registros, consulte Crear un grupo de CloudWatch registros en la Guía del usuario de Logs.
- Amazon S3: usted especifica el bucket de S3 al que quiere que MSK Connect envíe los eventos de registro de su conector. Para obtener más información sobre la creación de un bucket de S3, consulte Crear un bucket en la Guía del usuario de Amazon S3.
- Amazon Data Firehose: especifica el flujo de entrega al que quiere que MSK Connect envíe los eventos de registro de su conector. Para obtener información sobre la creación de un flujo de entrega, consulte Creación de un flujo de entrega de Amazon Data Firehose en la Guía del usuario de Firehose.

Para obtener más información sobre la configuración del registro, consulte Habilitar el registro desde determinados servicios de AWS en la Guía del usuario de Amazon CloudWatch Logs .

MSK Connect emite los siguientes tipos de eventos de registro:

Nivel	Descripción
`INFO`	Eventos de tiempo de ejecución de interés durante el inicio y el cierre.
`WARN`	Situaciones de tiempo de ejecución que no son errores pero que son indeseables o inesperadas.
`FATAL`	Errores graves que provocan una terminación prematura.
`ERROR`	Condiciones inesperadas y errores de tiempo de ejecución que no son irrecuperables.

A continuación, se muestra un ejemplo de un evento de registro enviado a CloudWatch Logs:

```
[Worker-0bb8afa0b01391c41] [2021-09-06 16:02:54,151] WARN [Producer
 clientId=producer-1] Connection to node 1 (b-1.my-test-cluster.twwhtj.c2.kafka.us-
east-1.amazonaws.com/INTERNAL_IP) could not be established. Broker may not be
 available. (org.apache.kafka.clients.NetworkClient:782)
```

Impedir que los secretos aparezcan en los registros de los conectores

> **Note**
>
> Los valores de configuración confidenciales pueden aparecer en los registros de los conectores si un complemento no los define como secretos. Kafka Connect trata los valores de configuración indefinidos de la misma manera que cualquier otro valor de texto sin formato.

Si su complemento define una propiedad como secreta, Kafka Connect redacta el valor de la propiedad de los registros del conector. Por ejemplo, los siguientes registros de conectores demuestran que si un complemento define `aws.secret.key` como un tipo `PASSWORD`, su valor se sustituye por **`[hidden]`**.

```
   2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b] [2022-01-11
15:18:55,150] INFO SecretsManagerConfigProviderConfig values:
   2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b] aws.access.key =
my_access_key
   2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b] aws.region = us-east-1
   2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b] aws.secret.key
= [hidden]
   2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b] secret.prefix =
   2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b] secret.ttl.ms = 300000
   2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b]
(com.github.jcustenborder.kafka.config.aws.SecretsManagerConfigProviderConfig:361)
```

Para evitar que los secretos aparezcan en los archivos de registro de los conectores, el desarrollador de un complemento debe usar la constante de enumeración de Kafka Connect `ConfigDef.Type.PASSWORD` para definir las propiedades confidenciales. Cuando una propiedad es del tipo `ConfigDef.Type.PASSWORD`, Kafka Connect excluye su valor de los registros del conector incluso si el valor se envía como texto sin formato.

Supervisión de MSK Connect

La supervisión es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de MSK Connect y sus demás AWS soluciones. Amazon CloudWatch monitorea tus AWS recursos y las aplicaciones en las que AWS ejecutas en tiempo real. Puede recopilar métricas y realizar un seguimiento de las métricas, crear paneles personalizados y definir alarmas que le advierten o que toman medidas cuando una métrica determinada alcanza el umbral que se especifique. Por ejemplo, puede CloudWatch hacer un seguimiento del uso de la CPU u otras métricas de su conector, de modo que pueda aumentar su capacidad si es necesario. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.

Puede utilizar las siguientes operaciones de API:

- `DescribeConnectorOperation`: Supervise el estado de las operaciones de actualización del conector.

- `ListConnectorOperations`: Realice un seguimiento de las actualizaciones anteriores ejecutadas en su conector.

En la siguiente tabla, se muestran las métricas a las que MSK Connect envía CloudWatch en la `ConnectorName` dimensión. MSK Connect proporciona estas métricas de forma predeterminada y sin coste adicional. CloudWatch conserva estas métricas durante 15 meses, para que pueda acceder a la información histórica y obtener una mejor perspectiva del rendimiento de sus conectores. También puede establecer alarmas que vigilen determinados umbrales y enviar notificaciones o realizar acciones cuando se cumplan dichos umbrales. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.

Métricas de MSK Connect

Nombre de métrica	Descripción
`BytesInPerSec`	El número total de bytes recibidos por el conector.
`BytesOutPerSec`	El número total de bytes entregados por el conector.
`CpuUtilization`	El porcentaje de consumo de CPU por sistema y usuario.
`ErroredTaskCount`	El número de tareas que han producido errores.
`MemoryUtilization`	El porcentaje de la memoria total de una instancia de proceso de trabajo, no solo de la memoria en montón de la máquina virtual Java (JVM) que se utiliza actualmente. Por lo general, la JVM no devuelve memoria al sistema operativo. Por lo tanto, el tamaño de pila de JVM (MemoryUtilization) normalmente comienza con un tamaño de pila mínimo que aumenta gradualmente hasta un máximo estable de aproximadamente el 80-90%. El uso en montón de JVM puede aumentar o disminuir

Nombre de métrica	Descripción
	a medida que cambia el uso real de la memoria del conector.
`RebalanceCompletedTotal`	El número total de reequilibrados realizados por este conector.
`RebalanceTimeAvg`	El tiempo medio en milisegundos que tarda el conector en reequilibrarse.
`RebalanceTimeMax`	El tiempo máximo en milisegundos que tarda el conector en reequilibrarse.
`RebalanceTimeSinceLast`	El tiempo en milisegundos desde que este conector completó el reequilibrio más reciente.
`RunningTaskCount`	El número de tareas en ejecución en el conector.
`SinkRecordReadRate`	El número medio de registros leídos por segundo desde el clúster de Apache Kafka o Amazon MSK.
`SinkRecordSendRate`	El número medio por segundo de registros que se generan a partir de las transformaciones y se envían al destino. Este número no incluye los registros filtrados.
`SourceRecordPollRate`	El número medio por segundo de registros producidos o sondeados.
`SourceRecordWriteRate`	El número medio de salida de registros por segundo desde las transformaciones y escrituras en el clúster de Apache Kafka o Amazon MSK.

Nombre de métrica	Descripción
`TaskStartupAttemptsTotal`	El número total de intentos de inicio de tareas del conector. Puede usar esta métrica para identificar anomalías en los intentos de inicio de tareas.
`TaskStartupSuccessPercentage`	El porcentaje medio de inicios de tareas satisfactorios del conector. Puede usar esta métrica para identificar anomalías en los intentos de inicio de tareas.
`WorkerCount`	El número de procesos de trabajo que se están ejecutando en el conector.

Ejemplos de configuración de los recursos de Amazon MSK Connect

En esta sección se incluyen ejemplos que le ayudarán a configurar los recursos de Amazon MSK Connect, como los conectores y proveedores de configuración habituales de terceros.

Temas

- Configuración de un conector de recepción de Amazon S3
- Configurar el conector de colector EventBridge Kafka para MSK Connect
- Uso del conector de origen Debezium con un proveedor de configuración

Configuración de un conector de recepción de Amazon S3

En este ejemplo se muestra cómo utilizar el conector receptor Amazon S3 de Confluent y cómo AWS CLI crear un conector receptor Amazon S3 en MSK Connect.

1. Copie el siguiente JSON y péguelo en un nuevo archivo. Sustituya las cadenas de marcador de posición por valores que correspondan a la cadena de conexión de los servidores de arranque del clúster de Amazon MSK y a la subred y al grupo de seguridad del clúster. IDs Para obtener más información sobre cómo configurar un rol de ejecución del servicio, consulte the section called “Roles y políticas de IAM”.

```
{
    "connectorConfiguration": {
        "connector.class": "io.confluent.connect.s3.S3SinkConnector",
        "s3.region": "us-east-1",
        "format.class": "io.confluent.connect.s3.format.json.JsonFormat",
        "flush.size": "1",
        "schema.compatibility": "NONE",
        "topics": "my-test-topic",
        "tasks.max": "2",
        "partitioner.class":
 "io.confluent.connect.storage.partitioner.DefaultPartitioner",
        "storage.class": "io.confluent.connect.s3.storage.S3Storage",
        "s3.bucket.name": "amzn-s3-demo-bucket"
    },
    "connectorName": "example-S3-sink-connector",
    "kafkaCluster": {
        "apacheKafkaCluster": {
            "bootstrapServers": "<cluster-bootstrap-servers-string>",
            "vpc": {
                "subnets": [
                    "<cluster-subnet-1>",
                    "<cluster-subnet-2>",
                    "<cluster-subnet-3>"
                ],
                "securityGroups": ["<cluster-security-group-id>"]
            }
        }
    },
    "capacity": {
        "provisionedCapacity": {
            "mcuCount": 2,
            "workerCount": 4
        }
    },
    "kafkaConnectVersion": "2.7.1",
    "serviceExecutionRoleArn": "<arn-of-a-role-that-msk-connect-can-assume>",
    "plugins": [
        {
            "customPlugin": {
                "customPluginArn": "<arn-of-custom-plugin-that-contains-connector-
code>",
                "revision": 1
            }
```

```
        }
    ],
    "kafkaClusterEncryptionInTransit": {"encryptionType": "PLAINTEXT"},
    "kafkaClusterClientAuthentication": {"authenticationType": "NONE"}
}
```

2. Ejecute el siguiente AWS CLI comando en la carpeta en la que guardó el archivo JSON en el paso anterior.

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

El siguiente es un ejemplo del resultado que se obtiene al ejecutar el comando correctamente.

```
{
    "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"
}
```

Configurar el conector de colector EventBridge Kafka para MSK Connect

En este tema se muestra cómo configurar el conector colector EventBridge Kafka para MSK Connect. Este conector le permite enviar eventos desde su clúster de MSK a EventBridge los buses de eventos. En este tema se describe el proceso para crear los recursos necesarios y configurar el conector para permitir un flujo de datos fluido entre Kafka y. EventBridge

Temas

- Requisitos previos
- Configure los recursos necesarios para MSK Connect
- Creación del conector
- Envía mensajes a Kafka

Requisitos previos

Antes de implementar el conector, asegúrese de que dispone de los siguientes recursos:

- Clúster de Amazon MSK: un clúster de MSK activo para producir y consumir mensajes de Kafka.

- Amazon EventBridge Event Bus: un autobús de EventBridge eventos para recibir eventos relacionados con los temas de Kafka.
- Funciones de IAM: cree funciones de IAM con los permisos necesarios para MSK Connect y EventBridge el conector.
- Acceso a Internet público desde MSK Connect o desde un punto final EventBridge de interfaz de VPC creado en la VPC y la subred de su clúster de MSK. Esto le ayuda a evitar atravesar la Internet pública y a no necesitar puertas de enlace NAT.
- Una máquina cliente, como una EC2 instancia de Amazon o AWS CloudShell, para crear temas y enviar registros a Kafka.

Configure los recursos necesarios para MSK Connect

Cree un rol de IAM para el conector y, a continuación, cree el conector. También crea una EventBridge regla para filtrar los eventos de Kafka enviados al bus de EventBridge eventos.

Temas

- Función de IAM para el conector
- Una EventBridge regla para los eventos entrantes

Función de IAM para el conector

La función de IAM que asocie al conector debe tener el PutEventspermiso para permitir el envío de eventos a. EventBridge El siguiente ejemplo de política de IAM le otorga el permiso para enviar eventos a un bus de eventos denominado. `example-event-bus` Asegúrese de reemplazar el ARN del recurso en el siguiente ejemplo por el ARN del bus de eventos.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
      ],
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/example-event-bus"
    }
  ]
}
```

Además, debe asegurarse de que su función de IAM para el conector contenga la siguiente política de confianza.

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

Una EventBridge regla para los eventos entrantes

Las reglas se crean para hacer coincidir los eventos entrantes con los criterios de los datos del evento, lo que se conoce como patrón de eventos. Con un patrón de eventos, puede definir los criterios para filtrar los eventos entrantes y determinar qué eventos deben activar una regla en particular y, posteriormente, enrutarse a un destino designado. El siguiente ejemplo de un patrón de eventos coincide con los eventos de Kafka enviados al bus de EventBridge eventos.

```
{
  "detail": {
    "topic": ["msk-eventbridge-tutorial"]
  }
}
```

A continuación se muestra un ejemplo de un evento enviado desde Kafka para EventBridge utilizar el conector colector de Kafka.

```
{
  "version": "0",
  "id": "dbc1c73a-c51d-0c0e-ca61-ab9278974c57",
  "account": "123456789012",
  "time": "2025-03-26T10:15:00Z",
  "region": "us-east-1",
  "detail-type": "msk-eventbridge-tutorial",
  "source": "kafka-connect.msk-eventbridge-tutorial",
```

```
  "resources": [],
  "detail": {
    "topic": "msk-eventbridge-tutorial",
    "partition": 0,
    "offset": 0,
    "timestamp": 1742984100000,
    "timestampType": "CreateTime",
    "headers": [],
    "key": "order-1",
    "value": {
      "orderItems": [
        "item-1",
        "item-2"
      ],
      "orderCreatedTime": "Wed Mar 26 10:15:00 UTC 2025"
    }
  }
}
```

En la EventBridge consola, cree una regla en el bus de eventos con este patrón de ejemplo y especifique un destino, como un grupo de CloudWatch registros. La EventBridge consola configurará automáticamente la política de acceso necesaria para el grupo de CloudWatch registros.

Creación del conector

En la siguiente sección, creará e implementará el conector receptor de EventBridge Kafka mediante el AWS Management Console.

Temas

- Paso 1: Descargar el conector
- Paso 2: creación de un bucket de Amazon S3
- Paso 3: Crear un complemento en MSK Connect
- Paso 4: creación del conector

Paso 1: Descargar el conector

Descargue el EventBridge conector JAR más reciente de la página de GitHub versiones del conector EventBridge Kafka. Por ejemplo, para descargar la versión 1.4.1, elija el enlace del archivo JAR para descargar el conector. `kafka-eventbridge-sink-with-dependencies.jar` A continuación, guarde el archivo en la ubicación que prefiera de su máquina.

Paso 2: creación de un bucket de Amazon S3

1. Para almacenar el archivo JAR en Amazon S3 para usarlo con MSK Connect, abra y AWS Management Console, a continuación, elija Amazon S3.
2. En la consola de Amazon S3, elija Create bucket e introduzca un nombre de bucket único. Por ejemplo, **`amzn-s3-demo-bucket1-eb-connector`**.
3. Elija una región adecuada para su bucket de Amazon S3. Asegúrese de que coincida con la región en la que está desplegado su clúster de MSK.
4. En cuanto a la configuración de Bucket, mantenga las selecciones predeterminadas o ajústelas según sea necesario.
5. Elija Crear bucket
6. Cargue el archivo JAR en el bucket de Amazon S3.

Paso 3: Crear un complemento en MSK Connect

1. Abre y AWS Management Console, a continuación, navega hasta MSK Connect.
2. En el panel de navegación izquierdo, selecciona Plugins personalizados.
3. Selecciona Crear complemento y, a continuación, introduce un nombre de complemento. Por ejemplo, **`eventbridge-sink-plugin`**.
4. En Ubicación personalizada del complemento, pega la URL del objeto S3.
5. Agrega una descripción opcional para el complemento.
6. Elige Crear complemento.

Una vez creado el complemento, puede usarlo para configurar e implementar el conector EventBridge Kafka en MSK Connect.

Paso 4: creación del conector

Antes de crear el conector, se recomienda crear el tema de Kafka correspondiente para evitar errores en el conector. Para crear el tema, utilice su máquina cliente.

1. En el panel izquierdo de la consola de MSK, selecciona Conectores y, a continuación, selecciona Crear conector.
2. En la lista de complementos, seleccione eventbridge-sink-plugin y, a continuación, seleccione Siguiente.

3. Para el nombre del conector, escriba**`EventBridgeSink`**.
4. En la lista de clústeres, elija su clúster de MSK.
5. Copie la siguiente configuración para el conector y péguela en el campo de configuración del conector

   Sustituya los marcadores de posición en la siguiente configuración, según sea necesario.

   - Elimine `aws.eventbridge.endpoint.uri` si su clúster de MSK tiene acceso público a Internet.
   - Si solía PrivateLink conectarse de forma segura desde MSK a EventBridge, sustituya la parte de DNS posterior `https://` por el nombre de DNS privado correcto del punto final de la interfaz de VPC (opcional) que EventBridge creó anteriormente.
   - Sustituya el ARN del bus de EventBridge eventos en la siguiente configuración por el ARN del bus de eventos.
   - Actualice cualquier valor específico de la región.

```
{
  "connector.class":
 "software.amazon.event.kafkaconnector.EventBridgeSinkConnector",
  "aws.eventbridge.connector.id": "msk-eventbridge-tutorial",
  "topics": "msk-eventbridge-tutorial",
  "tasks.max": "1",
  "aws.eventbridge.endpoint.uri": "https://events.us-east-1.amazonaws.com",
  "aws.eventbridge.eventbus.arn": "arn:aws:events:us-east-1:123456789012:event-bus/
example-event-bus",
  "value.converter.schemas.enable": "false",
  "value.converter": "org.apache.kafka.connect.json.JsonConverter",
  "aws.eventbridge.region": "us-east-1",
  "auto.offset.reset": "earliest",
  "key.converter": "org.apache.kafka.connect.storage.StringConverter"
}
```

   Para obtener más información sobre la configuración de los conectores, consulte. [eventbridge-kafka-connector]

   Si es necesario, cambie la configuración de los trabajadores y del escalado automático. También recomendamos utilizar la última versión disponible (recomendada) de Apache Kafka Connect del menú desplegable. En Permisos de acceso, utilice el rol creado anteriormente.

También recomendamos habilitar el registro en CloudWatch para facilitar la observación y solucionar problemas. Ajusta los demás ajustes opcionales, como las etiquetas, en función de tus necesidades. A continuación, despliegue el conector y espere a que el estado entre en ejecución.

Envía mensajes a Kafka

Puede configurar las codificaciones de los mensajes, como Apache Avro y JSON, especificando distintos convertidores mediante los `key.converter` ajustes `value.converter` y, opcionalmente, disponibles en Kafka Connect.

El contenido connector example de este tema está configurado para funcionar con mensajes codificados en JSON, como se indica mediante el uso de for. `org.apache.kafka.connect.json.JsonConverter` `value converter` Cuando el conector esté en estado de ejecución, envíe los registros al tema de `msk-eventbridge-tutorial` Kafka desde su máquina cliente.

Uso del conector de origen Debezium con un proveedor de configuración

En este ejemplo se muestra cómo utilizar el complemento de conector Debezium MySQL con una base de datos de Amazon Aurora compatible con MySQL como origen. En este ejemplo, también configuramos el proveedor de código abierto AWS Secrets Manager Config Provider para externalizar las credenciales de la base de datos en AWS Secrets Manager. Para obtener más información sobre los proveedores de configuración, consulte Tutorial: Externalización de la información confidencial con proveedores de configuración.

> **Important**
>
> El complemento de conector Debezium MySQL solo admite una tarea y no funciona con el modo de capacidad con escalado automático para Amazon MSK Connect. En su lugar, debe utilizar el modo de capacidad aprovisionada y establecer `workerCount` igual a uno en la configuración del conector. Para obtener más información sobre los modos de capacidad de MSK Connect, consulte Descripción de la capacidad de los conectores.

Requisitos previos que se deben completar para usar el conector de origen de Debezium

El conector debe poder acceder a Internet para poder interactuar con servicios como los AWS Secrets Manager que están fuera del suyo Amazon Virtual Private Cloud. Los pasos de esta sección le ayudan a realizar las siguientes tareas para habilitar el acceso a Internet.

- Configure una subred pública que aloje una puerta de enlace NAT y dirija el tráfico a una puerta de enlace de Internet en su VPC.
- Cree una ruta predeterminada que dirija el tráfico de su subred privada a su puerta de enlace NAT.

Para obtener más información, consulte Habilitación del acceso a Internet para Amazon MSK Connect.

Requisitos previos

Antes de habilitar el acceso a Internet, necesitará lo siguiente:

- El ID de la Amazon Virtual Private Cloud (VPC) asociada al clúster. Por ejemplo, vpc-123456ab.
- Las subredes privadas IDs de su VPC. Por ejemplo, subnet-a1b2c3de, subnet-f4g5h6ij, etc. Debe configurar el conector con subredes privadas.

Habilitación del acceso a Internet para su conector

1. Abra la Amazon Virtual Private Cloud consola en. https://console.aws.amazon.com/vpc/
2. Cree una subred pública para su puerta de enlace NAT con un nombre descriptivo y anote el ID de subred. Para obtener instrucciones detalladas, consulte Crear una subred en la VPC.
3. Cree una puerta de enlace de Internet para que su VPC pueda comunicarse con Internet y anote el ID de la puerta de enlace. Adjunte una puerta de enlace de Internet a su VPC. Para obtener más instrucciones, consulte Crear y adjuntar una puerta de enlace de Internet.
4. Aprovisione una puerta de enlace NAT pública para que los hosts de sus subredes privadas puedan acceder a su subred pública. Cuando cree la puerta de enlace NAT, seleccione la subred pública que creó anteriormente. Para obtener instrucciones, consulte Creación de una puerta de enlace NAT.
5. Configure sus tablas de enrutamiento. Debe tener dos tablas de enrutamiento en total para completar esta configuración. Ya debería tener una tabla de enrutamiento principal que se

haya creado automáticamente al mismo tiempo que su VPC. En este paso, crea una tabla de enrutamiento adicional para su subred pública.

a. Utilice la siguiente configuración para modificar la tabla de enrutamiento principal de la VPC de modo que las subredes privadas dirijan el tráfico a la puerta de enlace NAT. Para obtener instrucciones, consulte Trabajar con tablas de enrutamiento en la Guía del usuario de Amazon Virtual Private Cloud.

   Tabla de enrutamiento Private MSKC

Propiedad	Valor
Etiqueta de nombre	Le recomendamos que asigne a esta tabla de enrutamiento una etiqueta descriptiva con su nombre para ayudarlo a identific arla. Por ejemplo, Private MSKC.
Subredes asociadas	Sus subredes privadas
Una ruta para habilitar el acceso a Internet para MSK Connect	• Destino: 0.0.0.0/0 • Destino: su ID de puerta de enlace NAT. Por ejemplo, nat-12a345bc6789efg1h.
Una ruta local para el tráfico interno	• Destino: 10.0.0.0/16. Este valor puede variar en función del bloque CIDR de la VPC. • Objetivo: local

b. Siga las instrucciones en Crear de una tabla de enrutamiento personalizada para crear una para la subred pública. Al crear la tabla, introduzca un nombre descriptivo en el campo Etiqueta de nombre para ayudarle a identificar la subred a la que está asociada la tabla. Por ejemplo, Public MSKC.

c. Configure su tabla de enrutamiento Public MSKC con los siguientes ajustes.

Propiedad	Valor
Etiqueta de nombre	Public MSKC o un nombre descriptivo diferente que elija

Propiedad	Valor
Subredes asociadas	Su subred pública con puerta de enlace NAT
Una ruta para habilitar el acceso a Internet para MSK Connect	• Destino: 0.0.0.0/0 • Destino: su ID de puerta de enlace de Internet. Por ejemplo, igw-1a234bc5.
Una ruta local para el tráfico interno	• Destino: 10.0.0.0/16. Este valor puede variar en función del bloque CIDR de la VPC. • Objetivo: local

Ahora que ha habilitado el acceso a Internet para Amazon MSK Connect, está listo para crear un conector.

Creación de un conector de origen Debezium

En este procedimiento, se describe cómo crear un conector de origen Debezium.

1. Creación de un complemento personalizado

   a. Descargue la última versión estable del complemento MySQL Connector desde el sitio de Debezium. Anote la versión de lanzamiento de Debezium que haya descargado (la versión 2.x o la antigua serie 1.x). Más adelante en este procedimiento, creará un conector basado en su versión de Debezium.

   b. Descargue y extraiga el proveedor de configuración de AWS Secrets Manager.

   c. Coloque los siguientes archivos en el mismo directorio:

      • La carpeta `debezium-connector-mysql`
      • La carpeta `jcusten-border-kafka-config-provider-aws-0.1.1`

   d. Comprima el directorio que creó en el paso anterior en un archivo ZIP y, a continuación, cargue el archivo ZIP en un bucket de S3. Para obtener instrucciones, consulte Carga de objetos en la Guía del usuario de Amazon S3.

   e. Copie el siguiente JSON y péguelo en un archivo. Por ejemplo, `debezium-source-custom-plugin.json`. *<example-custom-plugin-name>*Sustitúyalo por el nombre

que desee que tenga el complemento, *<amzn-s3-demo-bucket-arn>* por el ARN del bucket de Amazon S3 en el que cargó el archivo ZIP y *<file-key-of-ZIP-object>* por la clave de archivo del objeto ZIP que cargó en S3.

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

f. Ejecute el siguiente AWS CLI comando desde la carpeta en la que guardó el archivo JSON para crear un complemento.

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

Debería ver un resultado similar a este.

```
{
    "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. Ejecute el siguiente comando para comprobar el estado del complemento. El estado debería cambiar de `CREATING` a `ACTIVE`. Sustituya el marcador de posición del ARN por el ARN que obtuvo en el resultado del comando anterior.

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

2. Configura AWS Secrets Manager y crea un secreto para las credenciales de tu base de datos

a. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.

b. Cree un nuevo secreto para almacenar sus credenciales de inicio de sesión de base de datos. Para obtener instrucciones, consulte Creación de un secreto en la Guía del usuario de AWS Secrets Manager.

c. Copie el ARN de su secreto.

d. Agregue los permisos de Secrets Manager desde la siguiente política de ejemplo a su Descripción del rol de ejecución de servicios. *<arn:aws:secretsmanager:us-east-1:123456789000:secret:MySecret-1234>*Sustitúyalo por el ARN de su secreto.

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

Para obtener información sobre cómo administrar los permisos de IAM, consulte Adición y eliminación de permisos de identidad de IAM en la Guía del usuario de IAM.

3. Creación de una configuración de proceso de trabajo personalizada con información sobre su proveedor de configuración

a. Copie las siguientes propiedades de configuración de proceso de trabajo en un archivo y sustituya las cadenas de marcadores de posición por valores que correspondan a su caso de uso. Para obtener más información sobre las propiedades de configuración del proveedor de configuración de AWS Secrets Manager, consulte SecretsManagerConfigProviderla documentación del complemento.

```
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. Ejecute el siguiente AWS CLI comando para crear su configuración de trabajo personalizada.

Reemplace los siguientes valores:

- *<my-worker-config-name>*- un nombre descriptivo para su configuración de trabajo personalizada
- *<encoded-properties-file-content-string>*- una versión codificada en base64 de las propiedades de texto sin formato que copió en el paso anterior

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

4. Creación de un conector

a. Copie el siguiente JSON que corresponda a su versión de Debezium (2.x o 1.x) y péguelo en un archivo nuevo. Sustituya las cadenas *<placeholder>* por valores que correspondan a su caso de uso. Para obtener más información sobre cómo configurar un rol de ejecución del servicio, consulte the section called “Roles y políticas de IAM”.

Tenga en cuenta que la configuración utiliza variables como `${secretManager:MySecret-1234:dbusername}` en lugar de texto sin formato para especificar las credenciales de la base de datos. Sustituya *MySecret-1234* por el nombre de su secreto y, a continuación, incluya el nombre de la clave que desea recuperar. También debe reemplazar *<arn-of-config-provider-worker-configuration>* por el ARN de su configuración de proceso de trabajo personalizada.

Debezium 2.x

Para las versiones 2.x de Debezium, copie el siguiente JSON y péguelo en un archivo nuevo. Sustituya las cadenas *<placeholder>* por valores que correspondan a su caso de uso.

```
{
```

```
 "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

Para las versiones 1.x de Debezium, copie el siguiente JSON y péguelo en un archivo nuevo. Sustituya las cadenas *<placeholder>* por valores que correspondan a su caso de uso.

```
{
  "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. Ejecute el siguiente AWS CLI comando en la carpeta en la que guardó el archivo JSON en el paso anterior.

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

El siguiente es un ejemplo del resultado que se obtiene al ejecutar el comando correctamente.

```
{
    "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"
}
```

Actualice la configuración de un conector Debezium

Para actualizar la configuración del conector Debezium, siga estos pasos:

1. Copie el siguiente JSON y péguelo en un archivo nuevo. Sustituya las cadenas `<placeholder>` por valores que correspondan a su caso de uso.

```
{
   "connectorArn": <connector_arn>,
```

```
   "connectorConfiguration": <new_configuration_in_json>,
   "currentVersion": <current_version>
}
```

2. Ejecuta el siguiente AWS CLI comando en la carpeta en la que guardaste el archivo JSON en el paso anterior.

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

A continuación, se muestra un ejemplo del resultado que se obtiene al ejecutar el comando correctamente.

```
{
    "connectorArn": "arn:aws:kafkaconnect:us-east-1:123450006789:connector/example-
Debezium-source-connector/abc12345-abcd-4444-a8b9-123456f513ed-2",
    "connectorOperationArn": "arn:aws:kafkaconnect:us-
east-1:123450006789:connector-operation/example-Debezium-source-connector/abc12345-
abcd-4444-a8b9-123456f513ed-2/41b6ad56-3184-479b-850a-a8bedd5a02f3",
    "connectorState": "UPDATING"
}
```

3. Ahora puede ejecutar el siguiente comando para supervisar el estado actual de la operación:

```
aws kafkaconnect describe-connector-operation --connector-operation-arn
 <operation_arn>
```

Para ver un ejemplo de conector Debezium con pasos detallados, consulte Introducing Amazon MSK Connect - Stream Data to and from Your Apache Kafka Clusters Using Managed Connectors.

Migración a Amazon MSK Connect

En esta sección, se describe cómo migrar su aplicación del conector de Apache Kafka a Amazon Managed Streaming para Apache Kafka Connect (Amazon MSK Connect). Para obtener más información sobre las ventajas de la migración a Amazon MSK Connect, consulte ???.

En esta sección, también se abordan los temas de administración de estados que utilizan Kafka Connect y Amazon MSK Connect, así como los procedimientos para migrar los conectores de origen y de recepción.

Descripción de los temas internos que utiliza Kafka Connect

Las aplicaciones de Apache Kafka Connect que se ejecutan en modo distribuido almacenan su estado mediante los temas internos en el clúster de Kafka y la pertenencia a grupos. A continuación se enumeran los valores de configuración correspondientes a los temas internos que se utilizan para las aplicaciones de Kafka Connect:

- Tema de configuración, especificado mediante `config.storage.topic`

  En el tema de configuración, Kafka Connect almacena la configuración de todos los conectores y las tareas que iniciaron los usuarios. Cada vez que los usuarios actualizan la configuración de un conector o cuando se requiere una reconfiguración en el conector (por ejemplo, el conector detecta que puede iniciar más tareas), se emite un registro a este tema. La compactación está habilitada para este tema, por lo que siempre mantiene el último estado de cada entidad.

- Tema de desplazamientos, especificado mediante `offset.storage.topic`

  En el tema de desplazamientos, Kafka Connect almacena los desplazamientos de los conectores de origen. Al igual que en el tema de configuración, la compactación está habilitada para el tema de desplazamientos. Este tema se utiliza para escribir las posiciones de origen únicamente para los conectores de origen que producen datos para Kafka desde sistemas externos. Los conectores de recepción, que leen datos de Kafka y los envían a sistemas externos, almacenan los desplazamientos de consumo en grupos de consumidores habituales de Kafka.

- Tema de estado, especificado mediante `status.storage.topic`

  Con el tema de estado, Kafka Connect almacena el estado actual de los conectores y las tareas. Este tema se utiliza como el punto central para los datos que consultan los usuarios de la API de REST. Con este tema, los usuarios pueden consultar cualquier proceso de trabajo y aún recibir el estado de todos los complementos en ejecución. Como en los temas de configuración y desplazamientos, la compactación también está habilitada para el tema de estado.

Además de estos temas, Kafka Connect utiliza ampliamente la API de pertenencia a grupos de Kafka. Los grupos se nombran como el conector. Por ejemplo, si el nombre de un conector es file-sink, el grupo se denomina. connect-file-sink Cada consumidor del grupo proporciona registros para una sola tarea. Estos grupos y sus desplazamientos se pueden recuperar con las herramientas para grupos de consumidores regulares, como `Kafka-consumer-group.sh`. Para cada conector de recepción, el tiempo de ejecución de Connect ejecuta un grupo de consumidores regular que extrae registros desde Kafka.

Administración del estado de las aplicaciones Amazon MSK Connect

De forma predeterminada, Amazon MSK Connect crea tres temas distintos para cada conector de Amazon MSK en el clúster de Kafka, con el fin de almacenar la configuración, los desplazamientos y el estado del conector. Los nombres predeterminados de los temas se estructuran de la siguiente manera:

- __msk_connect_configs_ _ *connector-name connector-id*
- __msk_connect_status_ _ *connector-name connector-id*
- __msk_connect_offsets_ _ *connector-name connector-id*

> **Note**
>
> Para proporcionar continuidad de desplazamiento entre los conectores de origen, puede utilizar el tema de almacenamiento de desplazamiento que desee en lugar del tema predeterminado. Especificar un tema de almacenamiento de desplazamiento lo ayuda a realizar tareas como crear un conector de origen que reanude la lectura desde el último desplazamiento de un conector anterior. Para especificar un tema de almacenamiento de desplazamiento, proporcione un valor para la propiedad offset.storage.topic en la configuración del proceso de trabajo Amazon MSK Connect antes de crear un conector.

Migración de los conectores de origen a Amazon MSK Connect

Los conectores de origen son aplicaciones de Apache Kafka Connect que importan registros desde sistemas externos a Kafka. En esta sección, se describe el proceso de migración de las aplicaciones de conectores de origen de Apache Kafka Connect que se ejecutan en las instalaciones o en los clústeres autoadministrados de Kafka Connect que se ejecutan en para AWS Amazon MSK Connect.

La aplicación de conectores de origen de Kafka Connect almacena los desplazamientos en un tema con el nombre del valor establecido en la propiedad de configuración `offset.storage.topic`. A continuación, se muestran ejemplos de mensajes de desplazamientos de un conector JDBC que ejecuta dos tareas para importar datos de dos tablas distintas, denominadas `movies` y `shows`. La última fila importada de la tabla denominada “películas” tiene un ID primario de `18343`. La última fila importada de la tabla denominada “programas” tiene un ID primario de `732`.

```
["jdbcsource",{"protocol":"1","table":"sample.movies"}] {"incrementing":18343}
```

```
["jdbcsource",{"protocol":"1","table":"sample.shows"}] {"incrementing":732}
```

Para migrar los conectores de origen a Amazon MSK Connect, siga los pasos a continuación:

1. Extrayendo las bibliotecas de conectores de sus clústeres en las instalaciones o autoadministrados de Kafta Connect, cree un complemento personalizado de Amazon MSK Connect.
2. Cree las propiedades del proceso de trabajo de Amazon MSK Connect y establezca las propiedades de `key.converter`, `value.converter` y `offset.storage.topic` en los mismos valores establecidos para el conector de Kafka que se ejecuta en su clúster existente de Kafka Connect.
3. Haga una solicitud de `PUT /connectors/`*`connector-name`*`/pause` en el clúster existente de Kafka Connect para pausar la aplicación de conectores en el clúster existente.
4. Asegúrese de que todas las tareas de la aplicación de conector estén completamente detenidas. Para detener las tareas, puede hacer una solicitud de `GET /connectors/`*`connector-name`*`/status` en el clúster existente de Kafka Connect o usar los mensajes del nombre de los temas establecido para la propiedad `status.storage.topic`.
5. Obtenga la configuración de los conectores del clúster existente. Para obtener la configuración de los conectores, puede hacer una solicitud de `GET /connectors/`*`connector-name`*`/config/` en el clúster existente o usar los mensajes del nombre de los temas establecido para la propiedad `config.storage.topic`.
6. Cree un nuevo conector de Amazon MSK con el mismo nombre que un clúster existente. Cree este conector usando el complemento personalizado de conectores que creó en el paso 1, las propiedades del proceso de trabajo que creó en el paso 2 y la configuración de los conectores que extrajo en el paso 5.
7. Cuando el estado del conector Amazon MSK sea `active`, consulte los registros para verificar que el conector haya empezado a importar datos del sistema de origen.
8. Haga una solicitud de `DELETE /connectors/`*`connector-name`* para eliminar el conector en el clúster existente.

Migración de los conectores de recepción a Amazon MSK Connect

Los conectores de recepción son aplicaciones de Apache Kafka Connect que exportan datos desde Kafka a sistemas externos. En esta sección, se describe el proceso de migración de las aplicaciones

de conectores de recepción de Apache Kafka Connect que se ejecutan en las instalaciones o en los clústeres autoadministrados de Kafka Connect que se ejecutan en para AWS Amazon MSK Connect.

Los conectores de recepción de Kafka Connect utilizan la API de pertenencia a grupos de Kafka y almacenan desplazamientos en los mismos temas de `__consumer_offset` que las aplicaciones de consumidor habituales. Este comportamiento simplifica la migración del conector de recepción de un clúster autoadministrado a Amazon MSK Connect.

Para migrar los conectores de recepción a Amazon MSK Connect, siga los pasos a continuación:

1. Extrayendo las bibliotecas de conectores de sus clústeres en las instalaciones o autoadministrados de Kafta Connect, cree un complemento personalizado de Amazon MSK Connect.
2. Cree las propiedades del proceso de trabajo de Amazon MSK Connect y establezca las propiedades de `key.converter` y `value.converter` en los mismos valores establecidos para el conector de Kafka que se ejecuta en su clúster existente de Kafka Connect.
3. Haga una solicitud de `PUT /connectors/`*`connector-name`*`/pause` en su clúster existente de Kafka Connect para pausar la aplicación de conectores en el clúster existente.
4. Asegúrese de que todas las tareas de la aplicación de conector estén completamente detenidas. Para detener las tareas, puede hacer una solicitud de `GET /connectors/`*`connector-name`*`/status` en el clúster existente de Kafka Connect o usar los mensajes del nombre de los temas establecido para la propiedad `status.storage.topic`.
5. Obtenga la configuración de conectores del clúster existente. Para obtener la configuración de los conectores, puede hacer una solicitud de `GET /connectors/`*`connector-name`*`/config` en el clúster existente o usar los mensajes del nombre de los temas establecido para la propiedad `config.storage.topic`.
6. Cree un nuevo conector de Amazon MSK con el mismo nombre que el clúster existente. Cree este conector usando el complemento personalizado de conectores que creó en el paso 1, las propiedades del proceso de trabajo que creó en el paso 2 y la configuración de los conectores que extrajo en el paso 5.
7. Cuando el estado del conector Amazon MSK sea `active`, consulte los registros para verificar que el conector haya empezado a importar datos del sistema de origen.
8. Haga una solicitud de `DELETE /connectors/`*`connector-name`* para eliminar el conector en el clúster existente.

Solución de problemas de Amazon MSK Connect

La siguiente información le puede ayudar a solucionar los problemas que podrían presentarse con MSK Connect. También puede publicar el problema en AWS re:Post.

Connector no puede acceder a los recursos alojados de forma pública en Internet

Consulte Habilitación del acceso a Internet para Amazon MSK Connect.

El número de tareas en ejecución de Connector no es igual al número de tareas especificadas en tasks.max

Estas son algunas de las razones por las que un conector puede usar menos tareas que la configuración de tasks.max especificada:

- Algunas implementaciones de conectores limitan la cantidad de tareas que se pueden utilizar. Por ejemplo, el conector Debezium para MySQL se limita a utilizar una sola tarea.
- Cuando se utiliza el modo de capacidad con escalado automático, Amazon MSK Connect anula la propiedad tasks.max de un conector con un valor que es proporcional al número de trabajadores que trabajan en el conector y al número de trabajadores por trabajador. MCUs
- En el caso de los conectores de recepción, el nivel de paralelismo (número de tareas) no puede ser superior al número de particiones temáticas. Si bien puede establecer el valor tasks.max en un tamaño mayor que ese valor, una sola partición nunca es procesada por más de una tarea a la vez.
- En Kafka Connect 2.7.x, el asignador de particiones de consumo predeterminado es `RangeAssignor`. El comportamiento de este asignador consiste en entregar la primera partición de cada tema a un solo consumidor, la segunda partición de cada tema a un solo consumidor, etc. Esto significa que el número máximo de tareas activas de un conector de recepción con `RangeAssignor` es igual al número máximo de particiones consumidas en un solo tema. Si esto no funciona para su caso de uso, debería crear una configuración de trabajo en la que la propiedad `consumer.partition.assignment.strategy` esté establecida en un asignador de particiones de consumo más adecuado. Consulte Interfaz Kafka 2.7: todas las clases de implementación conocidas. ConsumerPartitionAssignor

¿Qué es el Replicador Amazon MSK?

El Replicador Amazon MSK es una característica de Amazon MSK que le permite replicar los datos de forma fiable en clústeres de Amazon MSK situados en la misma AWS región de o en regiones diferentes. Con el Replicador MSK, puede crear fácilmente aplicaciones de streaming resistentes a nivel regional para aumentar la disponibilidad y la continuidad empresarial. El Replicador MSK proporciona una replicación asíncrona automática en los clústeres de MSK, lo que elimina la necesidad de escribir código personalizado, administrar la infraestructura o configurar redes entre regiones.

El Replicador MSK escala automáticamente los recursos subyacentes, para que pueda replicar los datos bajo demanda sin tener que supervisar ni escalar la capacidad. El Replicador MSK también replica los metadatos de Kafka necesarios, incluidas las configuraciones de los temas, las listas de control de acceso (ACLs) y los desplazamientos por grupos de consumidores. Si se produce un suceso inesperado en una región, puede hacer la conmutación por error a la otra AWS región de y reanudar el procesamiento de forma fluida.

El Replicador MSK admite la replicación entre regiones (CRR) y la replicación en la misma región (SRR). En la replicación entre regiones, los clústeres de MSK de origen y destino se encuentran en regiones de diferentes AWS . En la replicación en la misma región de, los clústeres de MSK de origen y destino se encuentran en la misma región de. AWS Debe crear clústeres de MSK de origen y de destino antes de usarlos con el Replicador MSK.

> **Note**
>
> MSK Replicador es compatible con las siguientes AWS regiones: EE. UU. Este (us-East-1, Virginia del Norte); EE. UU. Este (US-East-2, Ohio); EE. UU.); Asia Pacífico (ap-southeast-2, Sídney), Europa (eu-north-1, Estocolmo), Asia Pacífico (ap-south-1, Bombay), Europa (eu-west-1, Bombay), Europa (eu-west-3, Sídney), Europa (ap-south-1, Bombay), Europa (eu-west-2, Londres), Asia Pacífico (ap-northeast-1, Tokio), EE.UU. Oeste (us-west-1, norte de California), Norte de California), Canadá (ca-central-1).

Estos son algunos usos comunes del Replicador Amazon MSK.

- Creación de aplicaciones de streaming multirregionales: cree aplicaciones de streaming de alta disponibilidad y tolerantes a errores para aumentar la resistencia sin necesidad de configurar soluciones personalizadas.

- Acceso a los datos con menor latencia: proporcione acceso a los datos con menor latencia a los consumidores de diferentes regiones geográficas.
- Distribución de los datos entre los socios: copie los datos de un clúster de Apache Kafka a varios clústeres de Apache Kafka, para que los diferentes equipos o socios tengan sus propias copias de los datos.
- Adición de datos para análisis: copie los datos de varios clústeres de Apache Kafka en un solo clúster para generar fácilmente información sobre los datos agregados en tiempo real.
- Escritura de forma local y acceso a los datos en todo el mundo: configure la replicación multiactiva para propagar automáticamente las escrituras realizadas en una AWS región de a otras regiones, a fin de proporcionar datos con una latencia y un costo más bajos.

Funcionamiento del Replicador Amazon MSK

Para empezar a utilizar el Replicador MSK, debe crear un nuevo replicador en la región de del clúster de destino. AWS El Replicador MSK copia automáticamente todos los datos del clúster de la AWS región de principal denominado origen al clúster de la región de destino denominado destino. Los clústeres de origen y destino pueden encontrarse en la misma región de o en AWS regiones de diferentes. Deberá crear el clúster de destino si aún no existe.

Al crear un replicador, el Replicador MSK implementa todos los recursos necesarios en la AWS región de del clúster de destino para optimizar la latencia de la replicación de datos. La latencia de la replicación varía en función de diversos factores, como la distancia de la red entre AWS las regiones de de de los clústeres de MSK, la capacidad de rendimiento de los clústeres de origen y destino y la cantidad de particiones de los clústeres de origen y destino. El Replicador MSK escala automáticamente los recursos subyacentes, para que pueda replicar los datos bajo demanda sin tener que supervisar ni escalar la capacidad.

Replicación de los datos

De forma predeterminada, el Replicador MSK copia todos los datos de forma asíncrona desde el último desplazamiento de las particiones de temas del clúster de origen al clúster de destino. Si la opción “Detectar y copiar temas nuevos” está activada, el Replicador MSK detecta y copia los nuevos temas o particiones de temas en el clúster de destino automáticamente. Sin embargo, el Replicador puede tardar hasta 30 segundos en detectar y crear los nuevos temas o particiones de temas en el clúster de destino. Cualquier mensaje generado en el tema de origen antes de que se creara el tema en el clúster de destino no se replicará. Como alternativa, puede configurar el Replicador durante

la creación para que inicie la replicación desde el primer desplazamiento en las particiones de los temas del clúster de origen si desea replicar los mensajes existentes sobre sus temas en el clúster de destino.

El Replicador MSK no almacena los datos. Los datos se consumen del clúster de origen, se almacenan en memoria intermedia y se escriben en el clúster de destino. El búfer se borra automáticamente cuando los datos se escriben correctamente o cuando se produce un error tras volver a intentarlo. Toda la comunicación y los datos entre el Replicador MSK y sus clústeres siempre se cifran durante el tránsito. Se capturan todas las llamadas a la API de MSK Replicator`DescribeClusterV2`, como`CreateTopic`,`DescribeTopicDynamicConfiguration`. AWS CloudTrail Los registros del agente de MSK también reflejarán lo mismo.

El Replicador MSK crea temas en el clúster de destino con un factor de replicación de 3. Si lo necesita, puede modificar el factor de replicación directamente en el clúster de destino.

Replicación de metadatos

El Replicador MSK también admite la copia de los metadatos del clúster de origen al clúster de destino. Los metadatos incluyen la configuración de los temas, las listas de control de acceso (ACLs) y los desplazamientos de los grupos de consumidores. Al igual que la replicación de datos, la replicación de metadatos también se realiza de forma asíncrona. Para mejorar el rendimiento, el Replicador MSK prioriza la replicación de datos sobre la replicación de metadatos.

La siguiente tabla es una lista de las listas de control de acceso (ACLs) que el Replicador MSK copia.

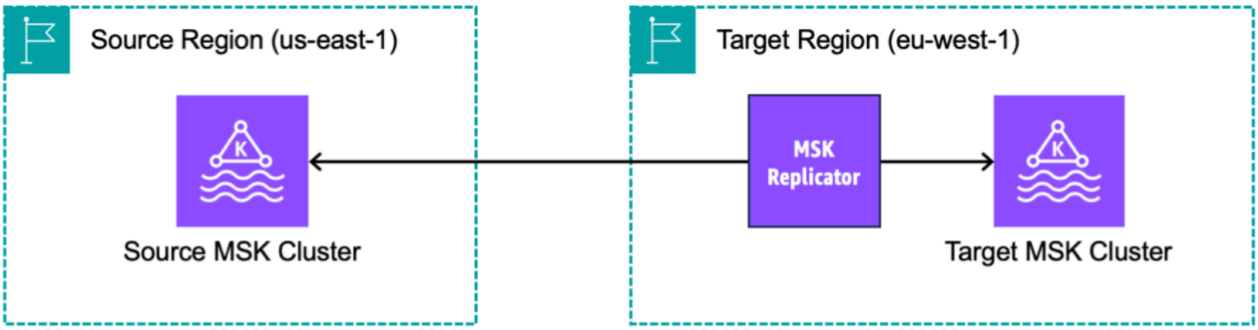
Operación	Investigación	APIs permitido
Modificación	Tema	CreatePartitions
AlterConfigs	Tema	AlterConfigs
Creación	Tema	CreateTopics, Metadatos
Eliminar	Tema	DeleteRecords, DeleteTopics
Describir	Tema	ListOffsets, Metadatos OffsetFetch, OffsetFor LeaderEpoch
DescribeConfigs	Tema	DescribeConfigs

Operación	Investigación	APIs permitido
Lectura	Tema	Buscar,, OffsetCommit TxnOffsetCommit
Escribir (solo denegar)	Tema	Producir, AddPartitionsToTxn

El Replicador MSK copia el tipo de patrón LITERAL ACLs solo para el tipo de recurso Topic. El tipo de patrón de PREFIJO ACLs y otros tipos de recursos no ACLs se copian. El Replicador MSK tampoco elimina del clúster ACLs de destino. Si elimina una ACL en el clúster de origen, también debe eliminarla en el clúster de destino al mismo tiempo. Para obtener más información sobre los ACLs recursos, el patrón y las operaciones de Kafka, consulte https://kafka.apache.org/documentation/#security_authz_cli.

El Replicador MSK solo replica a Kafka ACLs, que el control de acceso de IAM no utiliza. Si sus clientes utilizan el control de acceso de IAM para leer/escribir en sus clústeres de MSK, también debe configurar las políticas de IAM pertinentes en el clúster de destino para lograr una conmutación por error sin problemas. Esto también es válido tanto para las configuraciones de replicación de nombres de temas con prefijo como para los idénticos.

Como parte de la sincronización de desplazamiento entre grupos de consumidores, el Replicador MSK se optimiza para los consumidores del clúster de origen, que leen desde una posición más cercana a la punta de la transmisión (al final de la partición del tema). Si sus grupos de consumidores están rezagados en el clúster de origen, es posible que los grupos de consumidores del grupo de destino tengan un mayor retraso en comparación con los de origen. Esto significa que, tras la conmutación por error al clúster de destino, sus consumidores volverán a procesar más mensajes duplicados. Para reducir este retraso, los consumidores del clúster de origen tendrían que ponerse al día y empezar a consumir desde la punta de la transmisión (al final de la partición del tema). A medida que sus consumidores se pongan al día, el Replicador MSK reducirá automáticamente el retraso.



Configuración del nombre del tema

El Replicador MSK tiene dos modos de configuración de nombres de temas: replicación de nombres de tema con prefijo (predeterminado) o idéntico.

Replicación de nombres de temas con prefijo

De forma predeterminada, el Replicador MSK crea nuevos temas en el clúster de destino con un prefijo generado automáticamente que se agrega al nombre del tema del clúster de origen, como `<sourceKafkaClusterAlias>.topic`. Esto sirve para distinguir los temas replicados de otros temas del clúster de destino y para evitar que los datos se repliquen de manera circular entre los clústeres.

Por ejemplo, el Replicador MSK replica los datos en un tema denominado «topic» del clúster de origen a un tema nuevo del clúster de destino denominado < alias>.topic. sourceKafkaCluster Puede encontrar el prefijo que se agregará a los nombres de los temas del clúster de destino en el campo sourceKafkaClusterAlias mediante la `DescribeReplicator` API o la página de detalles del Replicador de la consola de MSK. El prefijo del clúster de destino es < sourceKafkaCluster Alias>.

Para garantizar que los consumidores puedan reiniciar el procesamiento de manera fiable desde el clúster en espera, debe configurar los consumidores para que lean los datos de los temas mediante el operador comodín `.*`. Por ejemplo, sus consumidores necesitarían consumir mediante. `*topic1`en ambas AWS regiones. Este ejemplo también incluiría un tema como `footopic1`, así que ajuste el operador comodín según sus necesidades.

Debe utilizar el Replicador de MSK que agrega un prefijo cuando desee conservar los datos del Replicador en un tema independiente del clúster de destino, por ejemplo, en el caso de las configuraciones de clústeres activo-activo.

Replicación de nombre de tema idéntico

Como alternativa a la configuración predeterminada, el Replicador Amazon MSK le permite crear un Replicador con la replicación de temas establecida en la replicación de nombres de temas idénticos (Mantener el mismo nombre de tema en la consola). Puede crear un nuevo Replicador en la AWS región de en la que se encuentre el clúster de MSK de destino. Los temas replicados con el mismo nombre le permiten evitar tener que volver a configurar los clientes para que lean temas replicados.

La replicación de nombres de temas idénticos (Mantener el mismo nombre de tema en la consola) tiene las siguientes ventajas:

- Permite retener nombres de temas idénticos durante el proceso de replicación y, al mismo tiempo, evita automáticamente el riesgo de ciclos de replicación infinitos.
- Simplifica la configuración y el funcionamiento de arquitecturas de transmisión de varios clústeres, ya que puede evitar tener que volver a configurar los clientes para leer los temas replicados.
- En el caso de las arquitecturas de clústeres activas y pasivas, la funcionalidad de replicación de nombres de temas idénticos también agiliza el proceso de conmutación por error, ya que permite a las aplicaciones realizar la conmutación por error sin problemas a un clúster en espera sin necesidad de cambiar el nombre de los temas ni reconfigurar el cliente.
- Se puede usar para consolidar más fácilmente los datos de varios clústeres de MSK en un solo clúster para la agregación de datos o el análisis centralizado. Esto requiere que cree Replicadores independientes para cada clúster de origen y para el mismo clúster de destino.
- Puede agilizar la migración de datos de un clúster de MSK a otro replicando los datos en temas con nombres idénticos en el clúster de destino.

El Replicador Amazon MSK utiliza los encabezados de Kafka para evitar automáticamente que los datos se repliquen en el tema en el que se originaron, lo que elimina el riesgo de ciclos infinitos durante la replicación. Un encabezado es un par clave-valor que se puede incluir con la clave, el valor y la marca de tiempo de cada mensaje de Kafka. El Replicador MSK incorpora los identificadores del clúster de origen y del tema en el encabezado de cada registro que se está replicando. El Replicador MSK utiliza la información del encabezado para evitar bucles de replicación infinitos. Debe comprobar que sus clientes pueden leer los datos replicados según lo esperado.

Tutorial: Configuración de clústeres de origen y destino para el Replicador Amazon MSK

En este tutorial, se muestra cómo configurar un clúster de origen y un clúster de destino en la misma AWS región de o en AWS regiones de diferentes. A continuación, utilice esos clústeres para crear un Replicador Amazon MSK.

Preparación del clúster de origen de Amazon MSK

Si ya tiene un clúster de origen de MSK para el Replicador MSK, asegúrese de que cumple los requisitos descritos en esta sección. De lo contrario, siga estos pasos para crear un clúster de origen aprovisionado o sin servidor de MSK.

El proceso para crear un clúster de origen del Replicador MSK entre regiones y en una misma región es similar. Las diferencias se indican en los procedimientos siguientes.

1. Cree un clúster aprovisionado o sin servidor de MSK con el control de acceso de IAM activado en la región de origen. El clúster de origen debe tener un mínimo de tres agentes.
2. En el caso de un Replicador MSK entre regiones, si el origen es un clúster aprovisionado, configúrelo con la conectividad privada de varias VPC activada para los esquemas de control de acceso de IAM. Tenga en cuenta que el tipo de autenticación no autenticada no se admite cuando varias VPC están activadas. No es necesario activar la conectividad privada de varias VPC para otros esquemas de autenticación (mTLS) ni esquemas de SASL/SCRAM). You can simultaneously use mTLS or SASL/SCRAM autenticación para sus otros clientes que se conectan al clúster de MSK. Puede configurar la conectividad privada de varias VPC en los detalles del clúster de la consola, en Configuración de red o con la API `UpdateConnectivity`. Consulte Cluster owner turns on multi-VPC. Si el clúster de origen es un clúster sin servidor de MSK, no es necesario activar la conectividad privada de varias VPC.

   Para un Replicador MSK de la misma región, el clúster de origen de MSK no requiere conectividad privada de varias VPC y otros clientes pueden seguir accediendo al clúster mediante el tipo de autenticación no autenticada.
3. En el caso de los replicadores de MSK entre regiones, debe asociar una política de permisos basada en recursos al clúster de origen. Esto permite a MSK conectarse a este clúster para replicar los datos. Para ello, puede utilizar los procedimientos de la CLI o de la AWS consola de que figuran a continuación. Consulte también la página sobre las políticas basadas en recursos

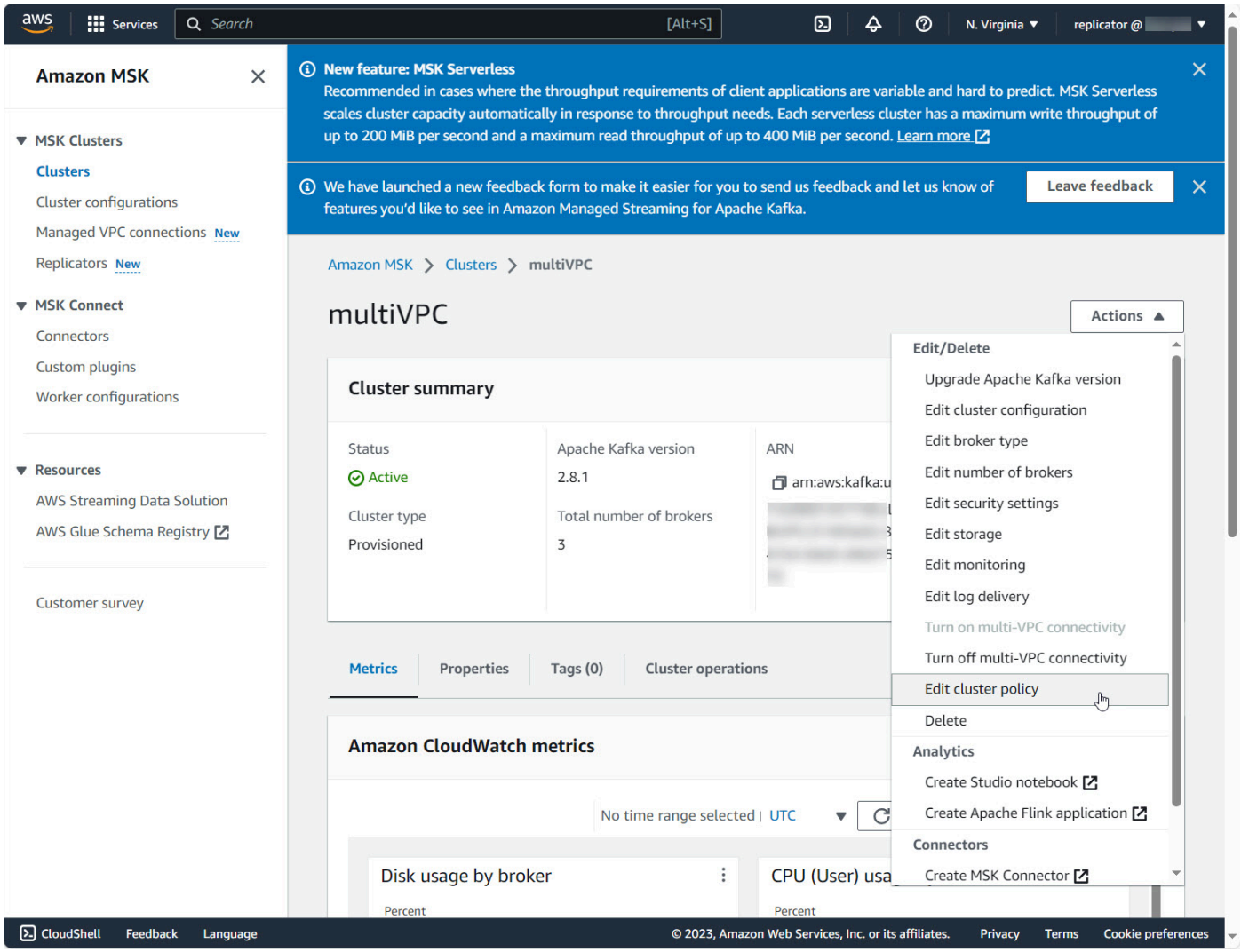
de Amazon MSK. No es necesario realizar este paso para los replicadores de MSK de la misma región.

Console: create resource policy

Actualice la política del clúster de origen con el siguiente elemento JSON. Sustituya el marcador de posición por el ARN del clúster de origen.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Service": [
                "kafka.amazonaws.com"
            ]
        },
        "Action": [
            "kafka:CreateVpcConnection",
            "kafka:GetBootstrapBrokers",
            "kafka:DescribeClusterV2"
        ],
        "Resource": "<sourceClusterARN>"
    }
  ]
}
```

Utilice la opción Editar la política del clúster en el menú Acciones de la página de detalles del clúster.



CLI: create resource policy

Nota: Si utiliza la AWS consola de para crear un clúster de origen y elige la opción para crear un nuevo rol de IAM, AWS adjunta la política de confianza necesaria al rol. Si quiere que MSK utilice un rol de IAM existente o si crea un rol por su cuenta, asocie las siguientes políticas de confianza a dicho rol para que MSK pueda asumirlo. Para obtener información acerca de cómo modificar la relación de confianza de un rol, consulte Modificación de un rol.

1. Obtenga la versión actual de la política de clústeres de MSK con este comando. Sustituya los marcadores de posición por el ARN del clúster real.

```
aws kafka get-cluster-policy –cluster-arn <Cluster ARN>
{
"CurrentVersion": "K1PA6795UKM GR7",
"Policy": "..."
```

```
}
```

2. Cree una política basada en recursos para permitir que el Replicador MSK acceda al clúster de origen. Utilice la siguiente sintaxis como plantilla y sustituya el marcador de posición por el ARN del clúster de origen real.

```
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>"
}
]
```

Preparación del clúster de destino de Amazon MSK

Cree un clúster de destino de MSK (aprovisionado o sin servidor) con el control de acceso de IAM activado. El clúster de destino no requiere que la conectividad privada de varias VPC esté activada. El clúster de destino puede encontrarse en la misma AWS región de o en una región diferente que el clúster de origen. Los clústeres de origen y destino deben estar en la misma AWS cuenta de. El clúster de destino debe tener un mínimo de tres agentes.

Tutorial: Creación de un Replicador Amazon MSK

Tras configurar los clústeres de origen y destino, puede utilizarlos para crear un Replicador Amazon MSK. Antes de crear el Replicador Amazon MSK, asegúrese de tener Permisos de IAM obligatorios para crear un Replicador MSK.

Temas

- Consideraciones sobre la creación de un Replicador Amazon MSK
  - Permisos de IAM obligatorios para crear un Replicador MSK
  - Tipos y versiones de clústeres compatibles para el Replicador MSK
  - Configuración de clústeres sin servidor de MSK compatibles
    - Cambios de configuraciones de clústeres
- Crear el replicador mediante la consola de AWS en la región del clúster de destino
  - Elección del clúster de origen
  - Elección del clúster de destino
  - Configurar los parámetros y los permisos del replicador

Consideraciones sobre la creación de un Replicador Amazon MSK

En las siguientes secciones se ofrece una descripción general de los requisitos previos, las configuraciones compatibles y las prácticas recomendadas para utilizar la característica del Replicador MSK. Abarca los permisos necesarios, la compatibilidad de los clústeres y los requisitos específicos sin servidor, así como orientación sobre la administración del Replicador tras su creación.

Permisos de IAM obligatorios para crear un Replicador MSK

Este es un ejemplo de la política de IAM necesaria para crear un Replicador MSK. La acción `kafka:TagResource` solo es necesaria si se proporcionan etiquetas al crear el Replicador MSK. Las políticas de IAM del Replicador se deben asociar al rol de IAM que corresponda al cliente. Para obtener información acerca de la creación de políticas de autorización, consulte Create authorization policies policies.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MSKReplicatorIAMPassRole",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/MSKReplicationRole",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "kafka.amazonaws.com"
        }
      }
```

```
    },
    {
      "Sid": "MSKReplicatorServiceLinkedRole",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::123456789012:role/aws-service-role/
kafka.amazonaws.com/AWSServiceRoleForKafka*"
    },
    {
      "Sid": "MSKReplicatorEC2Actions",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVpcs",
        "ec2:CreateNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:123456789012:subnet/subnet-0abcd1234ef56789",
        "arn:aws:ec2:us-east-1:123456789012:security-group/sg-0123abcd4567ef89",
        "arn:aws:ec2:us-east-1:123456789012:network-interface/eni-0a1b2c3d4e5f67890",
        "arn:aws:ec2:us-east-1:123456789012:vpc/vpc-0a1b2c3d4e5f67890"
      ]
    },
    {
      "Sid": "MSKReplicatorActions",
      "Effect": "Allow",
      "Action": [
        "kafka:CreateReplicator",
        "kafka:TagResource"
      ],
      "Resource": [
        "arn:aws:kafka:us-
east-1:123456789012:cluster/myCluster/abcd1234-56ef-78gh-90ij-klmnopqrstuv",
        "arn:aws:kafka:us-
east-1:123456789012:replicator/myReplicator/wxyz9876-54vu-32ts-10rq-ponmlkjihgfe"
      ]
    }
  ]
}
```

A continuación, se muestra un ejemplo de la política de IAM para describir el replicador. Se necesita la acción `kafka:DescribeReplicator` o la acción `kafka:ListTagsForResource`, no ambas.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "kafka:DescribeReplicator",
                "kafka:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

Tipos y versiones de clústeres compatibles para el Replicador MSK

Estos son requisitos para los tipos de instancias, las versiones de Kafka y las configuraciones de red compatibles.

- El Replicador MSK es compatible con clústeres aprovisionados de MSK y clústeres sin servidor de MSK en cualquier combinación como clústeres de origen y destino. Actualmente, el Replicador MSK no es compatible con otros tipos de clústeres de Kafka.
- Los clústeres sin servidor de MSK requieren el control de acceso de IAM, no son compatibles con la replicación de ACL de Apache Kafka y tienen una compatibilidad limitada para la replicación de configuraciones de temas. Consulte ¿Qué es MSK sin servidor?.
- El Replicador MSK solo es compatible con clústeres que ejecutan Apache Kafka 2.7.0 o superior, independientemente de si los clústeres de origen y destino se encuentran en el mismo lugar o en lugares diferentes de. Regiones de AWS
- El Replicador MSK es compatible con clústeres que utilizan tipos de instancia m5.large o superiores. No es compatible con los clústeres t3.small.
- Si utiliza el Replicador MSK con un clúster aprovisionado por MSK, se necesita un mínimo de tres agentes en los clústeres de origen y destino. Puede replicar datos en clústeres de dos zonas de disponibilidad, pero necesitará un mínimo de cuatro agentes en esos clústeres.
- Los clústeres de MSK de origen y destino deben estar en la misma AWS cuenta de. No se admite la replicación en clústeres de cuentas diferentes.

- Si los clústeres de MSK de origen y destino se encuentran en regiones de diferentes (entre AWS regiones), el Replicador MSK requiere que el clúster de origen tenga activada la conectividad privada de varias VPC para su método de control de acceso de IAM.

  No se requieren varias VPC para otros métodos de autenticación en el clúster de origen para la replicación de MSK en todo. Regiones de AWS

  Tampoco se requieren varias VPC si se replican datos entre clústeres de la misma. Región de AWS Consulte the section called “Conectividad privada con varias VPC en una sola región”.
- La replicación de nombres de temas idénticos (Mantener el mismo nombre de tema en la consola) requiere un clúster de MSK que ejecute la versión 2.8.1 o superior de Kafka.
- Para configuraciones de replicación de nombres de temas idénticos (Mantener el mismo nombre de tema en la consola), para evitar el riesgo de replicación cíclica, no realice cambios en los encabezados que crea el Replicador MSK (`__mskmr`).

Configuración de clústeres sin servidor de MSK compatibles

- MSK sin servidor es compatible con la replicación de estas configuraciones de temas para los clústeres de destino sin servidor de MSK durante la creación de los temas: `cleanup.policy`, `compression.type`, `max.message.bytes`, `retention.bytes` y `retention.ms`.
- MSK sin servidor solo es compatible con estas configuraciones de temas durante la sincronización de la configuración de temas: `compression.type`, `max.message.bytes`, `retention.bytes` y `retention.ms`.
- El replicador utiliza 83 particiones compactadas en los clústeres sin servidor de MSK de destino. Asegúrese de que los clústeres sin servidor de MSK de destino tengan suficientes particiones compactadas. Consulte Cuota de MSK sin servidor.

Cambios de configuraciones de clústeres

- Se recomienda no activar ni desactivar el almacenamiento por niveles una vez creado el Replicador MSK. Si el clúster de destino no está organizado por niveles, MSK no copiará las configuraciones de almacenamiento por niveles, independientemente de si el clúster de origen lo está o no. Si activa el almacenamiento por niveles en el clúster de destino después de crear el replicador, es necesario volver a crearlo. Si quiere copiar los datos de un clúster sin niveles a otro con niveles, no debe copiar las configuraciones de los temas. Consulte Habilitación y deshabilitación del almacenamiento por niveles en un tema existente.

- No cambie los parámetros de configuración del clúster después de crear el Replicador MSK. Los parámetros de configuración del clúster se validan durante la creación del Replicador MSK. Para evitar problemas con el Replicador MSK, no cambie los parámetros siguientes una vez creado el Replicador MSK.
  - Cambiar clúster de MSK al tipo de instancia t3.
  - Cambiar permisos de rol de ejecución de servicios.
  - Deshabilitar conectividad privada de varias VPC de MSK.
  - Cambiar política basada en recursos de clúster adjunto.
  - Cambiar reglas de grupos de seguridad de clúster.

Crear el replicador mediante la consola de AWS en la región del clúster de destino

En la siguiente sección se explica el flujo de trabajo de la consola paso a paso para crear un Replicador.

Detalles del Replicador

1. En la AWS región de en la que se encuentra el clúster de MSK de destino, ¿abre la consola de Amazon MSK en https://console.aws.amazon.com/msk/ casa? region=us-east-1#/home/.
2. Elija Replicadores para mostrar la lista de replicadores de la cuenta.
3. Elija Crear replicador.
4. En el panel Detalles del replicador, asigne un nombre único al nuevo replicador.

Elección del clúster de origen

El clúster de origen contiene los datos que quiere copiar a un clúster de MSK de destino.

1. En el panel Clúster de origen, elija la región de AWS en la que se encuentra el clúster de origen.

   Para buscar la región de un clúster, vaya a Clústeres de MSK y consulte el ARN de detalles del clúster. El nombre de la región está incrustado en la cadena de ARN. En el ejemplo siguiente de ARN, `ap-southeast-2` es la región del clúster.

   ```
   arn:aws:kafka:ap-southeast-2:123456789012:cluster/cluster-11/
   eec93c7f-4e8b-4baf-89fb-95de01ee639c-s1
   ```

2. Introduzca el ARN del clúster de origen o busque y elija el clúster de origen.
3. Elija las subredes para el clúster de origen.

   La consola muestra las subredes disponibles en la región del clúster de origen para que las seleccione. Debe seleccionar un mínimo de dos subredes. En el caso de un Replicador MSK de la misma región, las subredes que seleccione para acceder al clúster de origen y las subredes para acceder al clúster de destino deben encontrarse en la misma zona de disponibilidad.
4. Elija los grupos de seguridad para que el Replicador MSK acceda al clúster de origen.

   - Para la replicación entre regiones (CRR), no necesita proporcionar grupos de seguridad para el clúster de origen.
   - Para la replicación en la misma región (SRR), vaya a la EC2 consola de Amazon en https://console.aws.amazon.com/ec2/ y asegúrese de que los grupos de seguridad que proporcionará para el replicador tengan reglas de salida que permitan el tráfico hacia los grupos de seguridad del clúster de origen. Además, garantice que los grupos de seguridad del clúster de origen tengan reglas de entrada que permitan el tráfico desde los grupos de seguridad del Replicador proporcionados para el origen.

     Cómo agregar reglas de entrante al grupo de seguridad de su clúster:

     1. En la AWS consola de, seleccione el nombre del clúster para ir a los detalles del clúster de origen.
     2. Seleccione la pestaña Propiedades y, luego, desplácese hacia abajo hasta el panel Configuración de red para seleccionar el nombre del grupo de seguridad aplicado.
     3. Vaya a las reglas de entrada y seleccione Editar reglas de entrada.
     4. Seleccione Agregar regla.
     5. En la columna Tipo de la nueva regla, seleccione TCP personalizado.
     6. En la columna Rango de puertos, escriba 9098. El Replicador MSK utiliza el control de acceso de IAM para conectarse a su clúster, que utiliza el puerto 9098.
     7. En la columna Origen, escriba el nombre del grupo de seguridad que proporcionará durante la creación del Replicador para el clúster de origen (puede ser el mismo que el grupo de seguridad del clúster de origen de MSK) y, luego, seleccione Guardar reglas.

Cómo agregar reglas de salida al grupo de seguridad del Replicador proporcionado para el origen:

1. En la AWS consola de de Amazon EC2, vaya al grupo de seguridad que proporcionará durante la creación del Replicador para el origen.
2. Vaya a las reglas de salida y seleccione Editar reglas de salida.
3. Seleccione Agregar regla.
4. En la columna Tipo de la nueva regla, seleccione TCP personalizado.
5. En la columna Rango de puertos, escriba `9098`. El Replicador MSK utiliza el control de acceso de IAM para conectarse a su clúster, que utiliza el puerto 9098.
6. En la columna Origen, escriba el nombre del grupo de seguridad del clúster de origen de MSK y, a continuación, seleccione Guardar reglas.

> **Note**
>
> Como alternativa, si no desea restringir el tráfico mediante sus grupos de seguridad, puede agregar reglas de entrada y salida que permitan Todo el tráfico.
> 1. Seleccione Agregar regla.
> 2. En la columna Tipo, seleccione Todo el tráfico.
> 3. En la columna Origen, escriba `0.0.0.0/0` y, luego, seleccione Guardar reglas.

Elección del clúster de destino

El clúster de destino es el clúster aprovisionado o sin servidor de MSK en el que se copian los datos de origen.

> **Note**
>
> El Replicador MSK crea nuevos temas en el clúster de destino con un prefijo generado automáticamente que se agrega al nombre del tema. Por ejemplo, el Replicador MSK replica los datos en "`topic`" del clúster de origen a un tema nuevo del clúster de destino denominado `<sourceKafkaClusterAlias>.topic`. Esto sirve para distinguir los temas que contienen datos replicados del clúster de origen de otros temas del clúster de destino y para evitar que los datos se repliquen de manera circular entre los clústeres. Puede encontrar el prefijo que se agregará a los nombres de los temas del clúster de

destino en el campo sourceKafkaClusterAlias mediante la `DescribeReplicator` API o la página de detalles del Replicador de la consola de MSK. El prefijo del clúster de destino es `<sourceKafkaClusterAlias>`.

1. En el panel Clúster de destino, elija la AWS región de en la que se encuentra el clúster de destino.
2. Introduzca el ARN del clúster de destino o busque y elija el clúster de destino.
3. Elija las subredes para el clúster de destino.

   La consola muestra las subredes disponibles en la región del clúster de destino para que las seleccione. Seleccione un mínimo de dos subredes.
4. Elija los grupos de seguridad para que el Replicador MSK acceda al clúster de destino.

   Se muestran los grupos de seguridad disponibles en la región del clúster de destino para que los seleccione. El grupo de seguridad elegido se asocia a cada conexión. Para más información sobre el uso de los grupos de seguridad, consulte Controlar el tráfico hacia AWS los recursos de mediante grupos de seguridad en la Guía del usuario de Amazon VPC.

   - Tanto para la replicación entre regiones (CRR) como para la replicación en la misma región (SRR), vaya a la EC2 consola de Amazon en https://console.aws.amazon.com/ec2/y asegúrese de que los grupos de seguridad que le proporcionará al Replicador tengan reglas de salida que permitan el tráfico hacia los grupos de seguridad del clúster de destino. Además, asegúrese de que los grupos de seguridad del clúster de destino tengan reglas de entrada que acepten el tráfico de los grupos de seguridad del replicador proporcionados para el destino.

     Cómo agregar reglas de entrada al grupo de seguridad de su clúster:

     1. En la AWS consola de, seleccione el nombre del clúster para ir a los detalles del clúster de destino.
     2. Seleccione la pestaña Propiedades y, luego, desplácese hacia abajo hasta el panel Configuración de red para seleccionar el nombre del grupo de seguridad aplicado.
     3. Vaya a las reglas de entrada y seleccione Editar reglas de entrada.
     4. Seleccione Agregar regla.
     5. En la columna Tipo de la nueva regla, seleccione TCP personalizado.

6. En la columna Rango de puertos, escriba 9098. El Replicador MSK utiliza el control de acceso de IAM para conectarse a su clúster, que utiliza el puerto 9098.
7. En la columna Origen, escriba el nombre del grupo de seguridad que proporcionará durante la creación del Replicador para el clúster de destino (puede ser el mismo que el grupo de seguridad del clúster de MSK de destino) y, luego, seleccione Guardar reglas.

Cómo agregar reglas de salida al grupo de seguridad del Replicador proporcionado para el destino:

1. En la AWS consola de, vaya al grupo de seguridad que proporcionará durante la creación del Replicador para el destino.
2. Seleccione la pestaña Propiedades y, luego, desplácese hacia abajo hasta el panel Configuración de red para seleccionar el nombre del grupo de seguridad aplicado.
3. Vaya a las reglas de salida y seleccione Editar reglas de salida.
4. Seleccione Agregar regla.
5. En la columna Tipo de la nueva regla, seleccione TCP personalizado.
6. En la columna Rango de puertos, escriba 9098. El Replicador MSK utiliza el control de acceso de IAM para conectarse a su clúster, que utiliza el puerto 9098.
7. En la columna Origen, escriba el nombre del grupo de seguridad del clúster de destino de MSK y, a continuación, seleccione Guardar reglas.

> **Note**
>
> Como alternativa, si no desea restringir el tráfico mediante sus grupos de seguridad, puede agregar reglas de entrada y salida que permitan Todo el tráfico.
> 1. Seleccione Agregar regla.
> 2. En la columna Tipo, seleccione Todo el tráfico.
> 3. En la columna Origen, escriba `0.0.0.0/0` y, luego, seleccione Guardar reglas.

Configurar los parámetros y los permisos del replicador

1. En el panel Configuración del replicador, indique los temas que quiere replicar mediante expresiones regulares en las listas de permitidos y denegados. De manera predeterminada, se replican todos los temas.

> **Note**
>
> El Replicador MSK solo replica hasta 750 temas en orden. Si necesita replicar más temas, le recomendamos que cree un Replicador independiente. Vaya al Support Center de la AWS consola de y cree un caso de soporte si necesita soporte para más de 750 temas por replicador. Puede supervisar la cantidad de temas que se replican mediante la métrica TopicCount "». Consulte Cuota de bróker Amazon MSK Standard.

2. De forma predeterminada, el Replicador MSK inicia la replicación desde el último desplazamiento (el más reciente) de los temas seleccionados. Como alternativa, puede iniciar la replicación desde el desplazamiento más temprano (el más antiguo) de los temas seleccionados si desea replicar los datos existentes sobre los temas. Una vez creado el Replicador, no puede cambiar esta configuración. Esta configuración corresponde al `startingPosition`campo de la `CreateReplicator`solicitud y la `DescribeReplicator`respuesta APIs.
3. Elija una configuración de nombre de tema:

   - Replicación de nombres de temas `PREFIXED` (Agregar el prefijo al nombre de los temas en la consola): la configuración predeterminada. El Replicador MSK replica “topic1” del clúster de origen a un tema nuevo del clúster de destino con el nombre `<sourceKafkaClusterAlias>.topic1`.
   - Replicación de nombres de temas idénticos (Mantener el mismo nombre de tema en la consola): los temas del clúster de origen se replican con nombres de tema idénticos en el clúster de destino.

   Esta configuración corresponde al `TopicNameConfiguration` campo de la `CreateReplicator` solicitud y la `DescribeReplicator` respuesta APIs. Consulte Funcionamiento del Replicador Amazon MSK.

   > **Note**
   >
   > De forma predeterminada, el Replicador MSK crea nuevos temas en el clúster de destino con un prefijo generado automáticamente que se agrega al nombre del tema. Esto sirve para distinguir los temas que contienen datos replicados del clúster de origen de otros temas del clúster de destino y para evitar que los datos se repliquen de manera circular entre los clústeres. Como alternativa, puede crear un Replicador de MSK con una replicación de nombres de temas idénticos (Mantener el mismo nombre de tema en

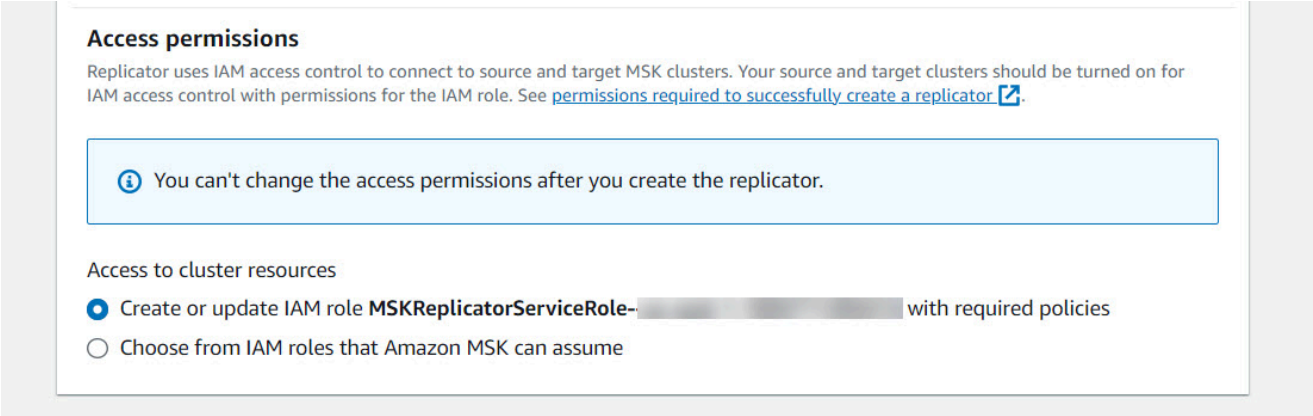
la consola) para que los nombres de los temas se conserven durante la replicación. Esta configuración reduce su necesidad de volver a configurar las aplicaciones de cliente durante la instalación y simplifica el funcionamiento de las arquitecturas de streaming de varios clústeres.

4. De manera predeterminada, el Replicador MSK copia todos los metadatos, incluidas las configuraciones de los temas, las listas de control de acceso (ACLs) y los desplazamientos de los grupos de consumidores para lograr una conmutación por error sin problemas. Si no va a crear el replicador para la conmutación por error, puede optar por desactivar una o varias de estas configuraciones disponibles en la sección Configuración adicional.

> **Note**
>
> El Replicador MSK no replica la escritura, ACLs ya que los productores no deberían escribir directamente en el tema replicado del clúster de destino. Tras la conmutación por error, los productores deberían escribir en el tema local del clúster de destino. Para obtener más información, consulte Realización de una conmutación por error planificada en la región secundaria AWS de.

5. En el panel Replicación del grupo de consumidores, indique los grupos de consumidores que quiere replicar mediante expresiones regulares en las listas de permitidos y denegados. De manera predeterminada, se replican todos los grupos de consumidores.
6. En el panel Compresión, si así lo quiere, puede optar por comprimir los datos escritos en el clúster de destino. Si va a utilizar la compresión, le recomendamos utilizar el mismo método de compresión que los datos del clúster de origen.
7. En el panel Permisos de acceso, realice uno de los siguientes procedimientos:

   a. Seleccione Crear o actualizar el rol de IAM con las políticas necesarias. La consola de MSK asociará automáticamente los permisos y la política de confianza necesarios al rol de ejecución del servicio necesario para leer los clústeres de MSK de origen y destino, y escribir en estos.



b. Proporcione su propio rol de IAM al seleccionar Elegir de los roles de IAM que puede asumir Amazon MSK. Le recomendamos que asocie la política de IAM administrada `AWSMSKReplicatorExecutionRole` a su rol de ejecución de servicios, en lugar de escribir su propia política de IAM.

   - Cree el rol de IAM que el Replicador utilizará para leer y escribir los clústeres de MSK de origen y destino con el siguiente JSON como parte de la política de confianza y el `AWSMSKReplicatorExecutionRole` asociado al rol. En la política de confianza, sustituya el marcador de posición <yourAccountID> por su ID de cuenta real.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "kafka.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<yourAccountID>"
                }
            }
        }
    ]
}
```

8. En el panel Etiquetas del Replicador, si así lo quiere, puede asignar etiquetas al recurso del Replicador MSK. Para obtener más información, consulte Etiquetado de un clúster de Amazon MSK. En el caso de un Replicador MSK entre regiones, las etiquetas se sincronizan

automáticamente con la región remota cuando se crea el replicador. Si se cambian las etiquetas después de crear el replicador, el cambio no se sincroniza automáticamente con la región remota, por lo que tendrá que sincronizar las referencias del replicador local y del replicador remoto de forma manual.

9. Seleccione Crear.

Si desea restringir los permisos `kafka-cluster:WriteData`, consulte la sección Create authorization policies de How IAM access control for Amazon MSK works. Deberá agregar permisos `kafka-cluster:WriteDataIdempotently` tanto al clúster de origen como al de destino.

El Replicador MSK tarda aproximadamente 30 minutos en crearse correctamente y pasar al estado EN EJECUCIÓN.

Si crea un nuevo Replicador MSK para reemplazar uno que haya eliminado, el nuevo replicador iniciará la replicación a partir del último desplazamiento.

Si el Replicador MSK ha pasado al estado ERROR, consulte la sección de solución de problemas del Replicador MSK.

Editar la configuración del Replicador MSK

No puede cambiar el clúster de origen, el clúster de destino, la posición de inicio del Replicador ni la configuración de replicación de nombres de temas una vez creado el Replicador MSK. Debe crear un Replicador nuevo para utilizar una configuración de replicación de nombres de temas idénticos. Sin embargo, puede editar otros ajustes del Replicador, como los temas y los grupos de consumidores que desee replicar.

1. ¿Iniciar sesión en la AWS Management Console y abrir la consola de Amazon MSK en https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. En el panel de navegación izquierdo, elija Replicadores para mostrar la lista de Replicadores de la cuenta y seleccione el Replicador MSK que quiere modificar.
3. Elija la pestaña Propiedades.
4. En la sección Configuración del replicador, elija Editar replicador.
5. Para editar la configuración del Replicador MSK, cambie cualquiera de estas configuraciones.

   - Indique los temas que quiere replicar mediante expresiones regulares en las listas de permitidos y denegados. De manera predeterminada, el Replicador MSK copia todos los

metadatos, incluidas las configuraciones de los temas, las listas de control de acceso (ACLs) y los desplazamientos de los grupos de consumidores para lograr una conmutación por error sin problemas. Si no va a crear el replicador para la conmutación por error, puede optar por desactivar una o varias de estas configuraciones disponibles en la sección Configuración adicional.

> **Note**
>
> El Replicador MSK no replica la escritura, ACLs ya que los productores no deberían escribir directamente en el tema replicado del clúster de destino. Tras la conmutación por error, los productores deberían escribir en el tema local del clúster de destino. Para obtener más información, consulte Realización de una conmutación por error planificada en la región secundaria AWS de.

- Para la replicación de grupos de consumidores, puede indicar los grupos de consumidores que quiere replicar mediante expresiones regulares en las listas de permitidos y denegados. De manera predeterminada, se replican todos los grupos de consumidores. Si las listas de permitidos y denegados están vacías, la replicación de grupos de consumidores se desactiva.
- En Tipo de compresión de destino, puede elegir si quiere comprimir los datos escritos en el clúster de destino. Si va a utilizar la compresión, le recomendamos utilizar el mismo método de compresión que los datos del clúster de origen.

6. Guarde los cambios.

   El Replicador MSK tarda aproximadamente 30 minutos en crearse correctamente y pasar al estado En ejecución. Si el Replicador MSK ha pasado al estado ERROR, consulte la sección de solución de problemas ???.

Eliminar un Replicador MSK

Es posible que tenga que eliminar un Replicador MSK si no se puede crear (estado ERROR). Los clústeres de origen y destino asignados a un Replicador MSK no se pueden cambiar una vez creado este. Puede eliminar un Replicador MSK existente y crear uno nuevo. Si crea un nuevo Replicador MSK para reemplazar el eliminado, el nuevo replicador iniciará la replicación a partir del último desplazamiento.

1. En la AWS región de en la que se encuentra el clúster de destino, ¿debe iniciar sesión en la AWS Management Console y abrir la consola de Amazon MSK en https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. En el panel de navegación, seleccione Replicadores.
3. En la lista de replicadores de MSK, seleccione el que quiere eliminar y elija Eliminar.

Supervisar la replicación

Puede utilizarlo https://console.aws.amazon.com/cloudwatch/en la región del clúster de destino para `ReplicationLatency` ver las métricas de cada Amazon MSK Replicator y `ReplicatorThroughput` a nivel de tema y agregado. `MessageLag` Las métricas están visibles ReplicatorNameen el espacio de nombres de «AWS/Kafka». También puede ver las métricas `ReplicatorFailure`, `AuthError` y `ThrottleTime` para comprobar si hay problemas.

La consola de MSK muestra un subconjunto de CloudWatch métricas para cada Replicador MSK. En la lista Replicadores de la consola, seleccione el nombre de un replicador y, luego, seleccione la pestaña Supervisión.

Métricas del Replicador MSK

Las siguientes métricas describen el rendimiento o conexión del Replicador MSK.

AuthError las métricas no cubren los errores de autenticación a nivel de temas. Para supervisar los errores de autenticación a nivel de tema del Replicador MSK, supervise las métricas del Replicador MSK y ReplicationLatency las métricas a nivel de tema del clúster de origen,. MessagesInPerSec Si un tema ReplicationLatency se reduce a 0, pero el tema aún contiene datos que se están produciendo, esto indica que el replicador tiene un problema de autenticación con el tema. Compruebe que el rol de IAM de ejecución de servicios del replicador tenga los permisos suficientes para acceder al tema.

Tipo de métrica	Métrica	Descripción	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar	
Rendimiento	ReplicationLatency	Tiempo que tardan en replicarse los registros desde el clúster de origen al de destino; tiempo transcurrido entre el tiempo de producción del registro en el origen y el tiempo de replicación en el de destino. Si ReplicationLatency aumenta, compruebe si los clústeres tienen particiones suficientes para admitir la replicación. Se puede producir una latencia de replicación alta cuando el	ReplicatorName	Milisegundos	Partición	Máximo	
			ReplicatorName, Tema	Milisegundos	Partición	Máximo	

Tipo de métrica	Métrica	Descripción	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar	
		recuento de particiones es demasiado bajo para un rendimiento alto.					

Tipo de métrica	Métrica	Descripción	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
Rendimiento	MessageLag	Supervisa la sincronización entre el Replicador MSK y el clúster de origen. MessageLag indica el desfase entre los mensajes producidos en el clúster de origen y los mensajes consumidos por el Replicador. No es el desfase entre el clúster de origen y el de destino. Incluso si el clúster de origen no está disponible/se interrumpe, el Replicador terminará de escribir el mensaje que ha consumido	ReplicatorName	Recuento	Partición	Sum
			ReplicatorName, Tema	Recuento	Partición	Sum

Tipo de métrica	Métrica	Descripción	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar	
		en el clúster de destino. Tras una interrupción, MessageLag muestra un aumento que indica el número de mensajes que el replicador está detrás del clúster de origen y se puede supervisar hasta que el número de mensajes sea 0, lo que indica que el replicador ha alcanzado el ritmo del clúster de origen.					

Tipo de métrica	Métrica	Descripción	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
Rendimiento	ReplicatorBytesInPerSec	El número promedio de bytes procesados por segundo por el Replicador. Los datos procesados por el Replicador MSK consisten en todos los datos que recibe el Replicador MSK, incluidos los datos replicados en el clúster de destino y los datos filtrados por el Replicador MSK (solo si el Replicador está configurado con nombres de temas idénticos) para evitar que los datos se copien de nuevo al mismo	ReplicatorName	BytesPerSecond	ReplicatorName	Sum

Tipo de métrica	Métrica	Descripción	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar	
		tema del que se originaron. Si su Replicador está configurado con nombres de temas con “Prefijo”, ambas métricas `ReplicatorBytesInPerSec` y `ReplicatorThroughput` tendrán el mismo valor, ya que el Replicador MSK no filtrará ningún dato.					

Tipo de métrica	Métrica	Descripción	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
Rendimiento	ReplicatorThroughput	El número medio de bytes replicados por segundo. Si ReplicatorThroughput disminuye para un tema, compruebe AuthError las métricas para garantizar que el replicador puede comunicarse con los clústeres KafkaClusterPingSuccessCount y, luego, compruebe las métricas del clúster para garantizar que el clúster no esté inactivo.	ReplicatorName	BytesPerSecond	Partición	Sum
			ReplicatorName, Tema	BytesPerSecond	Partición	Sum

Tipo de métrica	Métrica	Descripción	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
Debug	AuthError	El número de conexiones con una autenticación errónea por segundo. Si esta métrica es superior a 0, puede comprobar si la política de roles de ejecución de servicios del replicador es válida y asegurarse de que no se hayan establecido permisos de denegación para los permisos del clúster. En función de la dimensión ClusterAlias, puede identificar si el clúster de origen o de destino presenta	ReplicatorName, ClusterAlias	Recuento	Entorno de trabajo	Sum

Tipo de métrica	Métrica	Descripción	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
		errores de autenticación.				
Debug	ThrottleTime	El tiempo medio en ms que los agentes del clúster limitaron una solicitud. Establezca una limitación para evitar que el Replicador MSK sobrecargue el clúster. Si esta métrica es 0, el valor de replicationLatency no es alto y el valor de replicatorThroughput es el esperado, la limitación funciona según lo esperado. Si esta métrica es superior a 0, puede ajustar la limitación en consecuencia.	ReplicatorName, ClusterAlias	Milisegundos	Entorno de trabajo	Máximo

Tipo de métrica	Métrica	Descripción	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
Debug	ReplicatorFailure	Número de errores que sufre el replicador.	ReplicatorName	Recuento		Sum

Tipo de métrica	Métrica	Descripción	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
Debug	KafkaClusterPingSuccessCount	Indica el estado de la conexión del replicador al clúster de Kafka. Si este valor es 1, la conexión funciona correctamente. Si el valor es 0 o no hay ningún punto de datos, la conexión no funciona correctamente. Si el valor es 0, puede comprobar la configuración de los permisos de red o de IAM para el clúster de Kafka. En función de la dimensión ClusterAlias, puede identificar si esta métrica es para el clúster	ReplicatorName, ClusterAlias	Recuento		Sum

Tipo de métrica	Métrica	Descripción	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar	
		de origen o destino.					

Uso de la replicación para aumentar la resiliencia de una aplicación de streaming de Kafka en las regiones

Puede usar el Replicador MSK para configurar topologías de clústeres activo-activo o activo-pasivo a fin de aumentar la resistencia de la aplicación Apache Kafka en las regiones de. AWS En una configuración activo-activo, ambos clústeres de MSK prestan servicios activos de lectura y escritura. En una configuración activo-pasivo, solo un clúster de MSK a la vez ofrece datos de streaming de forma activa, mientras que el otro clúster se encuentra en espera.

Consideraciones para crear aplicaciones Apache Kafka de varias regiones

Los consumidores deben poder volver a procesar los mensajes duplicados sin que ello repercuta en las fases posteriores. El Replicador MSK replica los datos, at-least-once lo que puede provocar duplicados en el clúster en espera. Al cambiar a la AWS región secundaria de, es posible que los consumidores procesen los mismos datos más de una vez. El Replicador MSK prioriza la copia de datos por encima de los desplazamientos de los consumidores para mejorar el rendimiento. Tras una conmutación por error, el consumidor puede empezar a leer los desplazamientos anteriores, lo que resulta en un procesamiento duplicado.

Los productores y los consumidores también deben tolerar la pérdida de un mínimo de datos. Dado que el Replicador MSK replica los datos de forma asíncrona, cuando la AWS región principal de comienza a sufrir errores, no hay garantía de que todos los datos se repliquen en la región secundaria. Puede utilizar la latencia de replicación para determinar el número máximo de datos que no se copiaron en la región secundaria.

Uso de una topología de clúster activo-activo frente a una activo-pasivo

Una topología de clústeres activo-activo ofrece un tiempo de recuperación prácticamente nulo y permite que la aplicación de streaming funcione simultáneamente en varias regiones de AWS . Cuando un clúster de una región está dañado, las aplicaciones conectadas al clúster de la otra región siguen procesando los datos.

Las configuraciones activo-pasivo son adecuadas para aplicaciones que solo pueden ejecutarse en una región de AWS a la vez o cuando se necesita un mayor control sobre el orden de procesamiento de los datos. Las configuraciones activo-pasivo requieren más tiempo de recuperación que las activo-activo, ya que debe iniciar toda la configuración activo-pasivo, incluidos los productores y los consumidores, en la región secundaria para reanudar el flujo de datos tras una conmutación por error.

Creación de una configuración de clúster de Kafka activo-pasivo con las configuraciones recomendadas para los nombres de temas

Para una configuración activo-pasivo, recomendamos utilizar una configuración similar de productores, clústeres de MSK y consumidores (con el mismo nombre de grupo de consumidores) en dos regiones diferentes de. AWS Es importante que los dos clústeres de MSK tengan la misma capacidad de lectura y escritura para garantizar una replicación de datos fiable. Debe crear un Replicador MSK para copiar continuamente los datos del clúster principal al clúster en espera. También debe configurar los productores para que escriban los datos en los temas de un clúster de la misma AWS región de.

Para una configuración activa-pasiva, cree un nuevo Replicador con una replicación de nombres de temas idénticos (Mantener el mismo nombre de temas en la consola) para empezar a replicar los datos del clúster de MSK de la región principal a su clúster de la región secundaria. Le recomendamos que utilice un conjunto duplicado de productores y consumidores en las dos AWS regiones de, cada uno de los cuales se conecte al clúster de su propia región mediante su cadena de arranque. Esto simplifica el proceso de conmutación por error, ya que no requerirá cambios en la cadena de arranque. Para garantizar que los consumidores lean casi desde donde lo dejaron, los consumidores de los clústeres de origen y de destino deben tener el mismo ID de grupo de consumidores.

Si utiliza la replicación de nombres de temas idénticos (Mantener el mismo nombre de temas en la consola) para su Replicador MSK, este replicará sus temas con el mismo nombre que los temas de origen correspondientes.

Le recomendamos que configure los ajustes y los permisos a nivel de clúster para sus clientes en el clúster de destino. No es necesario configurar los ajustes a nivel de tema ni la lectura literal, ACLs ya que el Replicador MSK los copia automáticamente si ha seleccionado la opción de copiar las listas de control de acceso. Consulte Replicación de metadatos.

Conmutación por error a la región secundaria de AWS

Le recomendamos que supervise la latencia de replicación en la AWS región secundaria de mediante Amazon CloudWatch. Durante un evento de servicio en la AWS región principal de, es posible que la latencia de replicación aumente repentinamente. Si la latencia sigue aumentando, usa el AWS Service Health Dashboard para comprobar si hay eventos de servicio en la AWS región principal de. Si se produce un evento, puede hacer una conmutación por error a la AWS región secundaria de.

Realización de una conmutación por error planificada en la región secundaria AWS de

Puede hacer una conmutación por error planificada para probar la resistencia de la aplicación ante un suceso inesperado en la AWS región principal de donde se encuentra el clúster de MSK de origen. Una conmutación por error planificada no debería provocar la pérdida de datos.

Si utiliza una configuración de replicación de nombres de temas idénticos, siga estos pasos:

1. Cierre todos los productores y consumidores que se conectan al clúster de origen.
2. Cree un nuevo Replicador MSK para replicar los datos del clúster de MSK de la región secundaria al clúster de MSK de la región principal con replicación de nombres de temas idénticos (mantener el mismo nombre de tema en la consola). Esto es necesario para copiar los datos que escribirá en la región secundaria de nuevo en la región principal, de modo que pueda conmutar por recuperación a la región principal una vez finalizado el evento inesperado.
3. Inicie los productores y los consumidores que se conectan al clúster de destino de la AWS región secundaria de.

Si utiliza una configuración de nombres de temas con prefijo, siga estos pasos para realizar la conmutación por error:

1. Cierre todos los productores y consumidores que se conectan al clúster de origen.

2. Cree un nuevo Replicador MSK para replicar los datos del clúster de MSK de la región secundaria al clúster de MSK de la región principal. Esto es necesario para copiar los datos que escribirá en la región secundaria de nuevo en la región principal, de modo que pueda conmutar por recuperación a la región principal una vez finalizado el evento inesperado.
3. Inicie los productores del clúster de destino de la AWS región secundaria de.
4. En función de los requisitos de orden de mensajes de la aplicación, siga los pasos de una de las siguientes pestañas.

   No message ordering

   Si la aplicación no exige un orden de los mensajes, utilice un operador comodín para que los consumidores de la AWS región secundaria de lean tanto temas locales (por ejemplo, topic) como replicados (por ejemplo,`<sourceKafkaClusterAlias>.topic`) con un operador comodín (por ejemplo,`.*topic`).

   Message ordering

   Si la aplicación exige ordenar los mensajes, inicie los consumidores solo para los temas replicados del clúster de destino (por ejemplo, `<sourceKafkaClusterAlias>.topic`), pero no para los temas locales (por ejemplo, `topic`).
5. Espere a que todos los consumidores de los temas replicados del clúster de MSK de destino terminen de procesar todos los datos, de modo que el retraso entre consumidores sea 0 y el número de registros procesados también sea 0. A continuación, detenga los consumidores de los temas replicados del clúster de destino. En este punto, se han consumido todos los registros que se replicaron desde el clúster de MSK de origen al clúster de MSK de destino.
6. Inicie los consumidores para los temas locales (por ejemplo, `topic`) del clúster de MSK de destino.

Realización de una conmutación por error no planificada en la región secundaria de AWS

Puede hacer una conmutación por error no planificada cuando se produzca un evento de servicio en la AWS región principal de que tenga el clúster de MSK de origen y quiera redirigir temporalmente el tráfico a la región secundaria que tiene el clúster de MSK de destino. Una conmutación por error no planificada podría provocar la pérdida de algunos datos, ya que el Replicador MSK replica los datos de forma asíncrona. Puede realizar un seguimiento del retraso de los mensajes utilizando las métricas en ???.

Si utiliza una configuración de replicación de nombres de temas idénticos (Mantener el mismo nombre de tema en la consola), siga estos pasos:

1. Intente desactivar todos los productores y consumidores que se conectan al clúster de MSK de origen de la región principal. Es posible que esta operación no se realice correctamente debido a deficiencias en esa región.
2. Inicie los productores y consumidores que se conectan al clúster de MSK de destino en la AWS región secundaria de para completar la conmutación por error. Como el Replicador MSK también replica los metadatos, incluidas las desplazamientos por grupos de lectura ACLs y consumo, sus productores y consumidores reanudarán el procesamiento prácticamente desde donde lo dejaron antes de la conmutación por error.

Si utiliza la configuración de nombres de temas `PREFIX`, siga estos pasos para realizar la conmutación por error:

1. Intente desactivar todos los productores y consumidores que se conectan al clúster de MSK de origen de la región principal. Es posible que esta operación no se realice correctamente debido a deficiencias en esa región.
2. Inicie los productores y consumidores que se conectan al clúster de MSK de destino en la AWS región secundaria de para completar la conmutación por error. Como el Replicador MSK también replica los metadatos, incluidas las desplazamientos por grupos de lectura ACLs y consumo, sus productores y consumidores reanudarán el procesamiento prácticamente desde donde lo dejaron antes de la conmutación por error.
3. En función de los requisitos de orden de mensajes de la aplicación, siga los pasos de una de las siguientes pestañas.

   No message ordering

   Si la aplicación no exige un orden de los mensajes, utilice un operador comodín para que los consumidores de la AWS región de destino de lean tanto temas locales (por ejemplo`topic`) como replicados (por ejemplo`<sourceKafkaClusterAlias>.topic`) con un operador comodín (por ejemplo,`.*topic`).

   Message ordering

   1. Inicie los consumidores solo para los temas replicados del clúster de destino (por ejemplo, `<sourceKafkaClusterAlias>.topic`), pero no para los temas locales (por ejemplo, `topic`).

2. Espere a que todos los consumidores de los temas replicados del clúster de MSK de destino terminen de procesar todos los datos, de modo que el retraso del desplazamiento sea 0 y el número de registros procesados también sea 0. A continuación, detenga los consumidores de los temas replicados del clúster de destino. En este punto, se han consumido todos los registros que se replicaron desde el clúster de MSK de origen al clúster de MSK de destino.
3. Inicie los consumidores para los temas locales (por ejemplo, `topic`) del clúster de MSK de destino.

4. Una vez que el evento de servicio haya finalizado en la región principal, cree un nuevo Replicador MSK para replicar los datos del clúster de MSK de la región secundaria al clúster de MSK de la región principal con la posición de inicio del Replicador establecida como la más temprana. Esto es necesario para copiar los datos que escribirá en la región secundaria de nuevo en la región principal, de modo que pueda conmutar por recuperación a la región principal una vez finalizado el evento de servicio. Si no establece la posición de inicio del Replicador en la más temprana, cualquier dato que haya generado en el clúster de la región secundaria durante el evento de servicio en la región principal no se copiará de nuevo al clúster de la región principal.

Realización de una conmutación por recuperación a la región principal de AWS

Puede hacer una conmutación por recuperación a la AWS región principal de una vez finalizado el evento de servicio en esa región.

Si utiliza una configuración de replicación de nombres de temas idénticos, siga estos pasos:

1. Cree un nuevo Replicador MSK con el clúster secundario como origen y el clúster principal como destino; la posición inicial se establece como la más temprana y la replicación de nombres de temas idénticos (Mantener el mismo nombre de tema en la consola).

   Esto iniciará el proceso de copiar todos los datos escritos en el clúster secundario después de la conmutación por error a la región principal.
2. Supervise la `MessageLag` métrica en el nuevo Replicador en Amazon CloudWatch hasta que llegue`0`, lo que indica que todos los datos se han replicado del secundario al principal.
3. Una vez que se hayan replicado todos los datos, detenga la conexión de todos los productores al clúster secundario e inicie la conexión de los productores al clúster principal.

4. Espere a que la métrica `MaxOffsetLag` de los consumidores que se conectan al clúster secundario se conviertan en `0` para asegurarse de que han procesado todos los datos. Consulte Supervisión del desfase del consumidor.
5. Una vez que se hayan procesado todos los datos, detenga a los consumidores en la región secundaria e inicie la conexión de los consumidores al clúster principal para completar la conmutación por recuperación.
6. Elimine el Replicador que creó en el primer paso y que consiste en replicar los datos del clúster secundario al principal.
7. Compruebe que el Replicador existente que está copiando datos del clúster principal al secundario tenga el estado «EN EJECUCIÓN» y una `ReplicatorThroughput` métrica en Amazon CloudWatch `0`.

   Tenga en cuenta que, al crear un Replicador nuevo con la posición inicial de conmutación por recuperación más temprana, empezará a leer todos los datos de los temas de sus clústeres secundarios. En función de la configuración de retención de datos, es posible que los temas contengan datos procedentes del clúster de origen. Si bien el Replicador MSK filtra automáticamente esos mensajes, usted seguirá incurriendo en cargos por el procesamiento de datos y la transferencia de todos los datos de su clúster secundario. Puede realizar un seguimiento del total de datos procesados por el Replicador mediante `ReplicatorBytesInPerSec`. Consulte Métricas del Replicador MSK.

Si está utilizando una configuración de nombres de temas con prefijo, siga estos pasos:

Debe iniciar los pasos de conmutación por recuperación solo después de que la replicación del clúster de la región secundaria al clúster de la región secundaria funcione al mismo ritmo y la MessageLag métrica de Amazon CloudWatch se aproxime a 0. Una conmutación por recuperación planificada no debería provocar la pérdida de datos.

1. Desactive todos los productores y consumidores que se conectan al clúster de MSK de la región secundaria.
2. Para una topología activo-pasivo, elimine el replicador que replica los datos del clúster de la región secundaria a la región principal. No es necesario eliminar el replicador para una topología activo-activo.
3. Inicie los productores que se conectan al clúster de MSK de la región principal.
4. En función de los requisitos de orden de mensajes de la aplicación, siga los pasos de una de las siguientes pestañas.

No message ordering

Si la aplicación no exige un orden de los mensajes, utilice un operador comodín para que los consumidores de la AWS región principal de lean tanto temas locales (por ejemplo,`topic`) como replicados (por ejemplo,`<sourceKafkaClusterAlias>.topic`) con un operador comodín (por ejemplo,`.*topic`). Los consumidores de temas locales (por ejemplo, topic) se reanudarán desde el último desplazamiento que consumieron antes de la conmutación por error. Si había datos sin procesar de antes de la conmutación por error, se procesarán ahora. En el caso de una conmutación por error planificada, no debería existir dicho registro.

Message ordering

1. Inicie los consumidores solo para los temas replicados de la región principal (por ejemplo, `<sourceKafkaClusterAlias>.topic`), pero no para los temas locales (por ejemplo, `topic`).
2. Espere a que todos los consumidores de los temas replicados del clúster de la región principal terminen de procesar todos los datos, de modo que el retraso del desplazamiento sea 0 y el número de registros procesados también sea 0. A continuación, detenga los consumidores de los temas replicados del clúster de la región principal. En este momento, todos los registros que se produjeron en la región secundaria tras la conmutación por error se consumieron en la región principal.
3. Inicie los consumidores para los temas locales (por ejemplo, `topic`) del clúster de la región principal.

5. Compruebe que el Replicador existente, del clúster de la región principal al clúster de la región secundaria, tenga el estado EN EJECUCIÓN y funcione según lo esperado mediante las métricas y latencia `ReplicatorThroughput`.

Creación de una configuración activa-activa mediante el Replicador MSK

Si desea crear una configuración activa-activa en la que ambos clústeres de MSK entreguen lecturas y escrituras de forma activa, le recomendamos que utilice un Replicador de MSK con la replicación de nombres de temas con prefijo (Agregar el prefijo al nombre de los temas en la consola). Sin embargo, esto requerirá que reconfigure a sus consumidores para que lean los temas replicados.

Siga estos pasos para configurar la topología activo-activo entre el clúster de MSK de origen A y el clúster de MSK de destino B.

1. Cree un Replicador MSK con el clúster A de MSK como origen y el clúster B de MSK como destino.
2. Una vez que el Replicador MSK anterior se haya creado correctamente, cree un replicador con el clúster B como origen y el clúster A como destino.
3. Cree dos conjuntos de productores y que cada uno escriba datos al mismo tiempo en el tema local (por ejemplo, “topic”) del clúster de la misma región que el productor.
4. Cree dos grupos de consumidores y que cada uno lea los datos mediante una suscripción comodín (como». *topic») del clúster de MSK de la misma AWS región de que el consumidor. De esta manera, los consumidores leerán automáticamente los datos producidos localmente en la región del tema local (por ejemplo, `topic`), así como los datos replicados de otra región en el tema (con el prefijo `<sourceKafkaClusterAlias>.topic`). Estos dos grupos de consumidores deben tener un grupo de consumidores diferente IDs para que los desplazamientos de los grupos de consumidores no se sobrescriban cuando los copie el Replicador MSK en el otro clúster.

Si desea evitar la reconfiguración de sus clientes, en lugar de la replicación de nombres de temas con prefijo (Agregar el prefijo al nombre de los temas en la consola), puede crear Replicadores MSK mediante la replicación de nombres de temas idénticos (Mantener el mismo nombre de tema en la consola) para crear una configuración activa-activa. Sin embargo, tendrá que pagar cargos adicionales por procesamiento y transferencia de datos por cada Replicador. Esto se debe a que cada Replicador necesitará procesar el doble de la cantidad habitual de datos, uno para la replicación y otro para evitar bucles infinitos. Puede realizar un seguimiento de la cantidad total de datos procesados por cada Replicador mediante la métrica `ReplicatorBytesInPerSec`. Consulte Supervisar la replicación. Esta métrica incluye los datos replicados en el clúster de destino, así como los datos filtrados por el Replicador MSK para evitar que los datos se copien de nuevo al mismo tema del que se originaron.

> **Note**
>
> Si utiliza la replicación de nombres de temas idénticos (Mantener el mismo nombre de tema en la consola) para configurar la topología activa-activa, espere al menos 30 segundos después de eliminar un tema antes de volver a crear uno con el mismo nombre. Este periodo de espera ayuda a evitar que los mensajes duplicados se vuelvan a replicar en el clúster de origen. Los consumidores deben poder volver a procesar los mensajes duplicados sin que

ello repercuta en las fases posteriores. Consulte Consideraciones para crear aplicaciones Apache Kafka de varias regiones.

Migración de un clúster de Amazon MSK a otro mediante el Replicador MSK

Puede utilizar la replicación de nombres de temas idénticos para la migración de clústeres, pero sus consumidores deben poder procesar los mensajes duplicados sin que ello repercuta en las fases posteriores. Esto se debe a que el Replicador MSK proporciona la at-least-once replicación, lo que puede provocar la duplicación de mensajes en raras ocasiones. Si sus consumidores cumplen con este requisito, siga estos pasos.

1. Cree un Replicador que replique los datos del clúster anterior al clúster nuevo, con la posición inicial del Replicador establecida como la más temprana y mediante la replicación de nombres de temas idénticos (Mantener el mismo nombre de tema en la consola).
2. Configuración de los ajustes y los permisos a nivel de clúster en el nuevo clúster. No es necesario configurar los ajustes a nivel de tema ni la lectura «literal» ACLs, ya que el Replicador MSK los copia automáticamente.
3. Supervise la `MessageLag` métrica en Amazon CloudWatch hasta que llegue a 0, lo que indica que se han replicado todos los datos.
4. Una vez que se hayan replicado todos los datos, impida que los productores escriban datos en el clúster anterior.
5. Vuelva a configurar esos productores para que se conecten al nuevo clúster y se inicien.
6. Supervise la métrica `MaxOffsetLag` para que sus consumidores lean los datos del clúster anterior hasta convertirse en `0`, lo que indica que se han procesado todos los datos existentes.
7. Detener a los consumidores que se conectan al clúster anterior.
8. Reconfigure a los consumidores para que se conecten al nuevo clúster y se inicien.

Migración de una versión autoadministrada al MirrorMaker Replicador MSK

Para migrar de MirrorMaker (MM2) al Replicador MSK, siga estos pasos:

1. Detenga al productor que está escribiendo en su clúster de Amazon MSK de origen.
2. Permita MM2 replicar todos los mensajes sobre los temas de sus clústeres de origen. Puede supervisar el retraso de consumo del MM2 consumidor en su clúster de MSK de origen para determinar cuándo se han replicado todos los datos.
3. Cree un Replicador nuevo con la posición inicial establecida en última y la configuración de nombres de temas establecida como `IDENTICAL` (replicación de los mismos nombres de temas en la consola).
4. Una vez que el Replicador tenga el estado EN EJECUCIÓN, puede volver a iniciar a los productores a escribir en el clúster de origen.

Solución de problemas del Replicador MSK

La siguiente información puede ayudar a solucionar los problemas que podrían presentarse con el Replicador MSK. Consulte Solución de problemas del clúster de Amazon MSK para obtener información sobre la resolución de problemas sobre otras características de Amazon MSK. También puede publicar el problema en AWS re:Post.

El estado del Replicador MSK pasa de EN CREACIÓN a ERROR

Estas son algunas de las causas más comunes de los errores en la creación del Replicador MSK.

1. Compruebe que los grupos de seguridad proporcionados para la creación del replicador en la sección del clúster de destino tengan reglas de salida que permitan el tráfico a los grupos de seguridad del clúster de destino. Además, compruebe que los grupos de seguridad del clúster de destino tengan reglas de entrada que acepten el tráfico de los grupos de seguridad que proporcione para la creación del replicador en la sección del clúster de destino. Consulte Elección del clúster de destino.
2. Si va a crear un replicador para la replicación entre regiones, compruebe que el clúster de origen tenga activada la conectividad de varias VPC para el método de autenticación del control de acceso de IAM. Consulte Conectividad privada con varias VPC de Amazon MSK en una sola región. Compruebe también que la política de clústeres esté configurada en el clúster de origen, para que el Replicador MSK pueda conectarse al clúster de origen. Consulte Preparación del clúster de origen de Amazon MSK.
3. Compruebe que el rol de IAM que proporcionó durante la creación del Replicador MSK tiene los permisos necesarios para leer los clústeres de origen y destino, y para escribir en estos.

Compruebe también que el rol de IAM tenga permisos para escribir en los temas. Consulte Configurar los parámetros y los permisos del replicador

4. Compruebe que la red ACLs no bloquea la conexión entre el Replicador MSK y los clústeres de origen y destino.
5. Es posible que los clústeres de origen o destino no estén completamente disponibles cuando el Replicador MSK intente conectarse a ellos. Esto puede deberse a una carga, uso del disco o de la CPU excesivos, lo que hace que el replicador no pueda conectarse a los agentes. Solucione el problema con los agentes e intente crear el replicador de nuevo.

Tras hacer las validaciones anteriores, vuelva a crear el Replicador MSK.

El Replicador MSK aparece atascado en el estado EN CREACIÓN

A veces, la creación del Replicador MSK puede tardar hasta 30 minutos. Espere 30 minutos y compruebe de nuevo el estado del replicador.

El Replicador MSK no replica los datos o solo replica datos parciales

Siga estos pasos para solucionar los problemas de replicación de datos.

1. Compruebe que el replicador no esté ejecutando errores de autenticación mediante la AuthError métrica proporcionada por el Replicador MSK en Amazon. CloudWatch Si esta métrica es superior a 0, compruebe si la política del rol de IAM que proporcionó para el replicador es válida y que no se hayan establecido permisos de denegación para los permisos del clúster. En función de la dimensión ClusterAlias, puede identificar si el clúster de origen o de destino presenta errores de autenticación.
2. Compruebe que los clústeres de origen y destino no tengan ningún problema. Es posible que el replicador no pueda conectarse al clúster de origen o destino. Esto puede ocurrir debido a que hay demasiadas conexiones, el disco está al máximo de su capacidad o hay un uso elevado de la CPU.
3. Compruebe que se puede acceder a los clústeres de origen y destino desde el Replicador MSK mediante la métrica de Amazon KafkaClusterPingSuccessCount . CloudWatch En función de la dimensión ClusterAlias, puede identificar si el clúster de origen o de destino presenta errores de autenticación. Si el valor de esta métrica es 0 o no tiene ningún punto de datos, la conexión no funciona correctamente. Debe comprobar los permisos de la red y del rol de IAM que utiliza el Replicador MSK para conectarse a los clústeres.

4. Compruebe que el replicador no esté ejecutando errores debido a la falta de permisos a nivel de tema mediante la métrica ReplicatorFailure de Amazon. CloudWatch Si esta métrica es superior a 0, compruebe el rol de IAM que proporcionó para los permisos a nivel de tema.
5. Compruebe que la expresión regular que proporcionó en la lista de permitidos al crear el replicador coincide con los nombres de los temas que quiere replicar. Compruebe también que los temas no se excluyan de la replicación debido a una expresión regular de la lista de denegados.
6. Tenga en cuenta que el Replicador puede tardar hasta 30 segundos en detectar y crear los nuevos temas o particiones de temas en el clúster de destino. Cualquier mensaje generado en el tema de origen antes de que se creara el tema en el clúster de destino no se replicará si la posición inicial del Replicador es la última (opción predeterminada). Como alternativa, si desea replicar los mensajes existentes sobre sus temas en el clúster de destino, puede iniciar la replicación desde el primer desplazamiento de las particiones de temas del clúster de origen. Consulte Configurar los parámetros y los permisos del replicador.

Los desplazamientos de mensajes en el clúster de destino son diferentes a las del clúster de origen

Como parte de la replicación de datos, el Replicador MSK consume los mensajes del clúster de origen y los produce al clúster de destino. Esto puede provocar que los mensajes tengan diferentes desplazamientos en los clústeres de origen y destino. Sin embargo, si activó la sincronización de los desplazamientos de los grupos de consumidores durante la creación del Replicador, el Replicador MSK traducirá automáticamente los desplazamientos mientras copia los metadatos para que, tras la conmutación por error al clúster de destino, sus consumidores puedan reanudar el procesamiento casi desde donde lo dejaron en el clúster de origen.

El Replicador MSK no sincroniza los desplazamientos de los grupos de consumidores o el grupo de consumidores no existe en el clúster de destino

Siga estos pasos para solucionar los problemas de replicación de metadatos.

1. Compruebe que la replicación de los datos funciona según lo esperado. Si no es así, consulte El Replicador MSK no replica los datos o solo replica datos parciales.
2. Compruebe que la expresión regular que proporcionó en la lista de permitidos al crear el Replicador coincide con los nombres de los grupos de consumidores que quiere replicar. También compruebe que los grupos de consumidores no se excluyan de la replicación debido a una expresión regular de la lista de denegados.

3. Compruebe que el Replicador MSK haya creado el tema en el clúster de destino. El Replicador puede tardar hasta 30 segundos en detectar y crear los nuevos temas o particiones de temas en el clúster de destino. Cualquier mensaje generado en el tema de origen antes de que se creara el tema en el clúster de destino no se replicará si la posición inicial del Replicador es la última (opción predeterminada). Si su grupo de consumidores del clúster de origen solo ha consumido los mensajes que el Replicador MSK no ha replicado, el grupo de consumidores no se replicará en el clúster de destino. Una vez que el tema se haya creado correctamente en el clúster de destino, el Replicador MSK empezará a replicar los mensajes recién escritos en el clúster de origen al clúster de destino. Una vez que su grupo de consumidores comience a leer estos mensajes del origen, el Replicador MSK replicará automáticamente el grupo de consumidores en el clúster de destino. Como alternativa, si desea replicar los mensajes existentes sobre sus temas en el clúster de destino, puede iniciar la replicación desde el primer desplazamiento de las particiones de temas del clúster de origen. Consulte Configurar los parámetros y los permisos del replicador.

> **Note**
>
> El Replicador MSK optimiza la sincronización de los desplazamientos de los grupos de consumidores para los consumidores del clúster de origen, que leen desde una posición más cercana al final de la partición de temas. Si sus grupos de consumidores están rezagados en el clúster de origen, es posible que los grupos de consumidores del grupo de destino tengan un mayor retraso en comparación con los de origen. Esto significa que, tras la conmutación por error al clúster de destino, sus consumidores volverán a procesar más mensajes duplicados. Para reducir este retraso, los consumidores del clúster de origen tendrían que ponerse al día y empezar a consumir desde la punta de la transmisión (al final de la partición del tema). A medida que sus consumidores se pongan al día, el Replicador MSK reducirá automáticamente el retraso.

La latencia de replicación es alta o sigue aumentando

Estas son algunas de las causas comunes de la latencia alta de replicación.

1. Compruebe que tiene el número correcto de particiones en los clústeres de MSK de origen y destino. Tener muy pocas o demasiadas particiones puede afectar al rendimiento. Para instrucciones sobre cómo elegir el número de particiones, consulte Prácticas recomendadas para utilizar el Replicador MSK. La tabla siguiente muestra el número mínimo de particiones recomendado para obtener el rendimiento deseado con el Replicador MSK.

Rendimiento y número mínimo recomendado de particiones

Rendimiento (MB/s)	El número mínimo de particiones requerido
50	167
100	334
250	833
500	1666
1 000	3333

2. Compruebe que los clústeres de MSK de origen y destino tienen suficiente capacidad de lectura y escritura para admitir el tráfico de la replicación. El Replicador MSK actúa como consumidor del clúster de origen (salida) y como productor del clúster de destino (entrada). Por lo tanto, debe aprovisionar la capacidad del clúster para admitir el tráfico de la replicación, además del resto del tráfico de los clústeres. Consulte ??? para obtener orientación sobre el tamaño de los clústeres de MSK.
3. La latencia de la replicación puede variar para los clústeres de MSK en diferentes pares de AWS regiones de origen y destino, en función de la distancia geográfica entre los clústeres. Por ejemplo, la latencia de la replicación suele ser menor cuando se replica entre clústeres de las regiones de Europa (Irlanda) y Europa (Londres), en comparación con la replicación entre clústeres de las regiones de Europa (Irlanda) y Asia-Pacífico (Sídney).
4. Compruebe que el replicador no se vea limitado debido a las cuotas demasiado agresivas que se establezcan en los clústeres de origen o destino. Puede utilizar la ThrottleTime métrica proporcionada por el Replicador MSK en Amazon CloudWatch para ver el tiempo medio en milisegundos que los agentes del clúster de origen/destino limitaron una solicitud. Si esta métrica es superior a 0, debe ajustar las cuotas de Kafka para reducir las limitaciones, para que el replicador pueda recuperarse. Consulte Administración del rendimiento del Replicador MSK mediante cuotas de Kafka para obtener información sobre la administración de las cuotas de Kafka para el replicador.
5. ReplicationLatency y MessageLag podría aumentar cuando una AWS región de se degrada. Use el Panel de estado de servicio de AWS para comprobar si hay un evento de servicio de MSK en la región en la que se encuentra el clúster principal de MSK. Si se produce un evento de servicio, puede redirigir temporalmente las lecturas y escrituras de la aplicación a la otra región.

Solución de problemas de errores de MSK Replicator mediante métricas ReplicatorFailure

La ReplicatorFailure métrica le ayuda a monitorear y detectar problemas de replicación en MSK Replicator. Un valor distinto de cero de esta métrica suele indicar un problema de error de replicación, que puede deberse a los siguientes factores:

- Límites de tamaño del mensaje
- Infracciones de rango de marca de tiempo
- problemas de tamaño del lote

Si la ReplicatorFailure métrica indica un valor distinto de cero, sigue estos pasos para solucionar el problema.

> **Note**
>
> Para obtener más información sobre esta métrica, consulte Métricas del Replicador MSK.

1. Configure un cliente que pueda conectarse al clúster de MSK de destino y que tenga configuradas las herramientas CLI de Apache Kafka. Para obtener información acerca de la configuración del cliente y la herramienta CLI de Kafka, consulteConexión a un clúster de Amazon MSK.
2. ¿Abrir la consola de Amazon MSK en https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.

   A continuación, proceda del modo siguiente:

   a. Obtenga el MSK Replicator y el clúster de MSK de destino ARNs .
   b. Obtenga los puntos finales del broker del clúster de MSK de destino. Utilizará estos puntos de enlace en los pasos que se describen a continuación.
3. Antes de ejecutar los siguientes comandos, exporte el ARN del Replicador MSK y los puntos de conexión de agente que obtuvo en el paso anterior.

   Asegúrese de reemplazar los valores de marcador de posición de **<** *ReplicatorARN* **>**, **< >** y **<** *BootstrapServerString ConsumerConfigFile* **>** utilizados en los siguientes ejemplos por sus valores reales.

```
export TARGET_CLUSTER_SERVER_STRING=<BootstrapServerString>
```

```
export REPLICATOR_ARN=<ReplicatorARN>
```

```
export CONSUMER_CONFIG_FILE=<ConsumerConfigFile>
```

4. En el <*path-to-your-kafka-installation*>/bin directorio de, haga lo siguiente:

   a. Guarde el siguiente script y asígnele un nombre**query-replicator-failure-message.sh**.

```
#!/bin/bash

# Script: Query MSK Replicator Failure Message
# Description: This script queries exceptions from AWS MSK Replicator status
 topics
# It takes a replicator ARN and bootstrap server as input and searches for
 replicator exceptions
# in the replicator's status topic, formatting and displaying them in a
 readable manner
#
# Required Arguments:
#   --replicator-arn: The ARN of the AWS MSK Replicator
#   --bootstrap-server: The Kafka bootstrap server to connect to
#   --consumer.config: Consumer config properties file
# Usage Example:
#   ./query-replicator-failure-message.sh ./query-replicator-failure-message.sh
 --replicator-arn <replicator-arn> --bootstrap-server <bootstrap-server> --
consumer.config <consumer.config>

print_usage() {
  echo "USAGE: $0 ./query-replicator-failure-message.sh --replicator-arn
 <replicator-arn> --bootstrap-server <bootstrap-server> --consumer.config
 <consumer.config>"
  echo "--replicator-arn <String: MSK Replicator ARN>      REQUIRED: The ARN of
 AWS MSK Replicator."
  echo "--bootstrap-server <String: server to connect to>  REQUIRED: The Kafka
 server to connect to."
  echo "--consumer.config <String: config file>            REQUIRED: Consumer
 config properties file."
  exit 1
```

```
}

# Initialize variables
replicator_arn=""
bootstrap_server=""
consumer_config=""

# Parse arguments
while [[ $# -gt 0 ]]; do
  case "$1" in
    --replicator-arn)
      if [ -z "$2" ]; then
        echo "Error: --replicator-arn requires an argument."
        print_usage
      fi
      replicator_arn="$2"; shift 2 ;;
    --bootstrap-server)
      if [ -z "$2" ]; then
        echo "Error: --bootstrap-server requires an argument."
        print_usage
      fi
      bootstrap_server="$2"; shift 2 ;;
    --consumer.config)
      if [ -z "$2" ]; then
        echo "Error: --consumer.config requires an argument."
        print_usage
      fi
      consumer_config="$2"; shift 2 ;;
    *) echo "Unknown option: $1"; print_usage ;;
  esac
done

# Check for required arguments
if [ -z "$replicator_arn" ] || [ -z "$bootstrap_server" ] || [ -z
 "$consumer_config" ]; then
  echo "Error: --replicator-arn, --bootstrap-server, and --consumer.config are
 required."
  print_usage
fi

# Extract replicator name and suffix from ARN
replicator_arn_suffix=$(echo "$replicator_arn" | awk -F'/' '{print $NF}')
replicator_name=$(echo "$replicator_arn" | awk -F'/' '{print $(NF-1)}')
echo "Replicator name: $replicator_name"
```

```
# List topics and find the status topic
topics=$(./kafka-topics.sh --command-config client.properties --list --
bootstrap-server "$bootstrap_server")
status_topic_name="__amazon_msk_replicator_status_${replicator_name}_
${replicator_arn_suffix}"

# Check if the status topic exists
if echo "$topics" | grep -Fq "$status_topic_name"; then
  echo "Found replicator status topic: '$status_topic_name'"
  ./kafka-console-consumer.sh --bootstrap-server "$bootstrap_server" --
consumer.config "$consumer_config" --topic "$status_topic_name" --from-
beginning | stdbuf -oL grep "Exception" | stdbuf -oL sed -n 's/.*Exception:\(.*
\) Topic: \([^,]*\), Partition: \([^\]*\).*/ReplicatorException:\1 Topic: \2,
 Partition: \3/p'
else
  echo "No topic matching the pattern '$status_topic_name' found."
fi
```

b. Ejecute este script para consultar los mensajes de error de MSK Replicator.

```
<path-to-your-kafka-installation>/bin/query-replicator-failure-message.sh --
replicator-arn $REPLICATOR_ARN --bootstrap-server $TARGET_CLUSTER_SERVER_STRING
 --consumer.config $CONSUMER_CONFIG_FILE
```

Este script genera todos los errores con sus mensajes de excepción y las particiones temáticas afectadas. Puede utilizar esta información sobre excepciones para mitigar los errores tal y como se describe en. Fallos comunes de MSK Replicator y sus soluciones Como el tema contiene todos los mensajes de errores históricos, inicie la investigación utilizando el último mensaje. A continuación se muestra un ejemplo de un mensaje de error.

```
ReplicatorException: The request included a message larger than the max message
 size the server will accept. Topic: test, Partition: 1
```

Fallos comunes de MSK Replicator y sus soluciones

La siguiente lista describe algunos de los errores de MSK Replicator que puede experimentar y cómo mitigarlos.

Tamaño del mensaje superior a max.request.size

Causa

Este error se produce cuando el replicador de MSK no puede replicar los datos porque el tamaño del mensaje individual supera los 10 MB. De forma predeterminada, MSK Replicator replica los mensajes de hasta 10 MB de tamaño.

A continuación se muestra un ejemplo de este tipo de mensaje de error.

```
ReplicatorException: The message is 20635370 bytes when serialized which is larger
 than 10485760, which is the value of the max.request.size configuration. Topic:
 test, Partition: 1
```

Solución

Reduce el tamaño de los mensajes individuales de tu tema. Si no puedes hacerlo, sigue estas instrucciones para solicitar un aumento del límite.

Un tamaño de mensaje superior al tamaño máximo de mensaje que el servidor aceptará

Causa

Este error se produce cuando el tamaño del mensaje supera el tamaño máximo de mensaje del clúster de destino.

A continuación se muestra un ejemplo de este tipo de mensaje de error.

```
ReplicatorException: The request included a message larger than the max message size
 the server will accept. Topic: test, Partition: 1
```

Solución

Aumente la `max.message.bytes` configuración en el clúster de destino o en el tema del clúster de destino correspondiente. Establezca la `max.message.bytes` configuración del clúster de destino para que coincida con el tamaño más grande de los mensajes sin comprimir. Para obtener información sobre cómo hacerlo, consulta max.message.bytes.

La marca de tiempo está fuera del rango

Causa

Este error se produce porque la marca de tiempo del mensaje individual queda fuera del rango permitido por el clúster de destino.

A continuación se muestra un ejemplo de este tipo de mensaje de error.

```
ReplicatorException: Timestamp 1730137653724 of message with offset 0 is out of
 range. The timestamp should be within [1730137892239, 1731347492239] Topic: test,
 Partition: 1
```

Solución

Actualice la `message.timestamp.before.max.ms` configuración del clúster de destino para permitir los mensajes con marcas de tiempo antiguas. Para obtener información sobre cómo hacerlo, consulta message.timestamp.before.max.ms.

Registre un lote demasiado grande

Causa

Este error se produce porque el tamaño del lote de registros supera el tamaño de segmento establecido para el tema en el clúster de destino. El Replicador MSK admite un tamaño máximo de lote de 1 MB.

A continuación se muestra un ejemplo de este tipo de mensaje de error.

```
ReplicatorException: The request included message batch larger than the configured
 segment size on the server. Topic: test, Partition: 1
```

Solución

La configuración segment.bytes del clúster de destino debe ser al menos tan grande como el tamaño del lote (1 MB) para que Replicator funcione sin errores. Actualice el segment.bytes del clúster de destino para que tenga al menos 1048576 (1 MB). Para obtener información sobre cómo hacerlo, consulte segment.bytes.

> **Note**
>
> Si la ReplicatorFailure métrica sigue emitiendo valores distintos de cero después de aplicar estas soluciones, repita el proceso de solución de problemas hasta que la métrica emita un valor cero.

Prácticas recomendadas para utilizar el Replicador MSK

En esta sección, se describen las prácticas recomendadas y las estrategias de implementación más comunes para utilizar el Replicador Amazon MSK.

Temas

- Administración del rendimiento del Replicador MSK mediante cuotas de Kafka
- Establecimiento del periodo de retención del clúster

Administración del rendimiento del Replicador MSK mediante cuotas de Kafka

Dado que el Replicador MSK actúa como consumidor del clúster de origen, la replicación puede provocar que otros consumidores se vean limitados a utilizar el clúster de origen. El grado de limitación varía en función de la capacidad de lectura del clúster de origen y del rendimiento de los datos que replicará. Le recomendamos que aprovisione una capacidad idéntica para los clústeres de origen y de destino, y que tenga en cuenta el rendimiento de la replicación al calcular la capacidad que necesita.

También puede establecer cuotas de Kafka para el replicador en los clústeres de origen y de destino a fin de controlar la capacidad que puede utilizar el Replicador MSK. Se recomienda una cuota de ancho de banda de la red. Una cuota de ancho de banda de la red define un umbral de velocidad de bytes, definido como bytes por segundo, para uno o varios clientes que comparten una cuota. Esta cuota se define por agente.

Siga estos pasos para aplicar una cuota.

1. Recupere la cadena del servidor de arranque del clúster de origen. Consulte Obtención de agentes de arranque para un clúster de Amazon MSK.
2. Recupere el rol de ejecución de servicios (SER) que utiliza el Replicador MSK. Este es el SER que utilizó para una solicitud `CreateReplicator`. También puede extraer el SER de la DescribeReplicator respuesta de un replicador existente.
3. Con las herramientas de la CLI de Kafka, ejecute el siguiente comando en el clúster de origen.

```
./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. Tras ejecutar el comando anterior, compruebe que la métrica `ReplicatorThroughput` no supere la cuota que ha establecido.

Tenga en cuenta que, si reutiliza un rol de ejecución de servicios entre varios replicadores de MSK, todos estarán sujetos a esta cuota. Si quiere mantener cuotas independientes por replicador, utilice roles de ejecución de servicios independientes.

Para más información sobre el uso de la autenticación de IAM de MSK con cuotas, consulte Multi-tenancy Apache Kafka clusters in Amazon MSK with IAM access control and Kafka Quotas – Part 1.

> **Warning**
>
> Si se establece una tasa de consumo extremadamente baja, es posible que el Replicador MSK actúe de forma inesperada.

Establecimiento del periodo de retención del clúster

Puede establecer el periodo de retención de los registros para los clústeres aprovisionados y sin servidor de MSK. El periodo de retención recomendado es de 7 días. Consulte Cambios de configuraciones de clústeres o Configuración de clústeres sin servidor de MSK compatibles.

Integraciones de MSK

En esta sección se proporcionan referencias a AWS las funciones que se integran con Amazon MSK.

Temas

- Conector de Amazon Athena para Amazon MSK
- Ingesta de datos de transmisión de Amazon Redshift para Amazon MSK
- Integración de Firehose en Amazon MSK
- Acceda a Amazon EventBridge Pipes a través de la consola Amazon MSK
- Uso de Kafka Streams con los corredores de MSK Express y MSK Serverless
- Planos de incrustación vectorial en tiempo real

Conector de Amazon Athena para Amazon MSK

El conector de Amazon Athena para Amazon MSK permite que Amazon Athena ejecute consultas SQL en los temas de Apache Kafka. Use este conector para ver los temas de Apache Kafka como tablas y los mensajes como filas en Athena.

Para obtener más información, consulte la sección sobre el Conector de MSK de Amazon Athena en la Guía del usuario de Amazon Athena.

Ingesta de datos de transmisión de Amazon Redshift para Amazon MSK

Amazon Redshift admite la ingesta de streaming desde Amazon MSK. La característica de ingesta de streaming de Amazon Redshift proporciona una ingesta de alta velocidad y baja latencia de datos de streaming de datos de Amazon MSK en una vista materializada de Amazon Redshift. Como no necesita almacenar los datos en Amazon S3, Amazon Redshift puede ingerir datos de streaming con una latencia más baja y con un costo de almacenamiento reducido. Puede configurar la ingesta de streaming de Amazon Redshift en un clúster de Amazon Redshift mediante instrucciones SQL para autenticarse y conectarse a un tema de Amazon MSK.

Para obtener más información, consulte la sección sobre Ingesta de streaming en la Guía para desarrolladores de bases de datos de Amazon Redshift.

Integración de Firehose en Amazon MSK

Amazon MSK se integra en Firehose para ofrecer una solución sin servidor y sin código y entregar flujos desde los clústeres de Apache Kafka a los lagos de datos de Amazon S3. Firehose es un servicio de extracción, transformación y carga (ETL) de transmisión que lee datos de sus temas de Amazon MSK Kafka, realiza transformaciones, como la conversión a Parquet, y agrega y escribe los datos en Amazon S3. De manera sencilla desde la consola, puede configurar un flujo de Firehose para leer un tema de Kafka y entregarla en una ubicación de S3. No hay que escribir código, no hay aplicaciones de conexión ni recursos que aprovisionar. Firehose se escala automáticamente según la cantidad de datos publicados en el tema de Kafka, mientras que solo paga por los bytes ingeridos desde Kafka.

Consulte lo siguiente para obtener más información acerca de esta característica.

- Escritura en Kinesis Data Firehose con Amazon MSK: Amazon Kinesis Data Firehose en la Guía para desarrolladores de Amazon Data Firehose
- Blog: Amazon MSK Introduces Managed Data Delivery from Apache Kafka to Your Data Lake
- Laboratorio: Delivery to Amazon S3 using Amazon Data Firehose

Acceda a Amazon EventBridge Pipes a través de la consola Amazon MSK

Amazon EventBridge Pipes conecta las fuentes con los objetivos. Los tubos están diseñados para la point-to-point integración entre las fuentes y los objetivos compatibles, y permiten realizar transformaciones y enriquecimientos avanzados. EventBridge Los pipes proporcionan una forma altamente escalable de conectar su clúster de Amazon MSK a AWS servicios como Step Functions, Amazon SQS y API Gateway, así como a aplicaciones de software como servicio (SaaS) de terceros, como Salesforce.

Para configurar una canalización, elija el origen, agregue filtros opcionales, defina el enriquecimiento opcional y elija el destino de los datos del evento.

En la página de detalles de un clúster de Amazon MSK, puede ver las canalizaciones que utilizan ese clúster como origen. Desde allí, también puede hacer lo siguiente:

- Inicie la EventBridge consola para ver los detalles de Pipe.
- Inicie la EventBridge consola para crear una nueva tubería con el clúster como fuente.

Para obtener más información sobre la configuración de un clúster de Amazon MSK como fuente canalizada, consulte Amazon Managed Streaming for Apache Kafka Cluster as a source en la Guía del usuario de EventBridge Amazon. Para obtener más información sobre EventBridge Pipes en general, consulte EventBridge Pipes.

Para acceder a EventBridge las tuberías de un clúster de Amazon MSK determinado

1. Abra la consola de Amazon ECS y seleccione Clústeres.
2. Seleccione un clúster.
3. En la página de detalles del clúster, seleccione la pestaña Integración.

   La pestaña Integración incluye una lista de todas las canalizaciones actualmente configuradas para usar el clúster seleccionado como origen, que incluye:

   - nombre de la canalización
   - estado actual
   - destino de la canalización
   - cuándo se modificó la canalización por última vez
4. Administre las canalizaciones de su clúster de Amazon MSK como desee:

   Acceso a más detalles sobre una canalización

   - Elija la canalización.

     Esto abre la página de detalles de Pipe de la EventBridge consola.

   Creación de una nueva canalización

   - Elija Conectar clúster de Amazon MSK a canalización.

     Esto abre la página Crear canalización de la EventBridge consola, con el clúster Amazon MSK especificado como fuente de canalización. Para obtener más información, consulta Cómo crear una EventBridge tubería en la Guía del EventBridge usuario de Amazon.

   - También puede crear una canalización para un clúster desde la página Clústeres. Seleccione el clúster y, en el menú Acciones, seleccione Crear EventBridge tubería.

Uso de Kafka Streams con los corredores de MSK Express y MSK Serverless

Kafka Streams admite transformaciones sin estado y con estado. Las transformaciones con estado, como contar, agregar o unir, utilizan operadores que almacenan su estado en temas internos de Kafka. Además, algunas transformaciones sin estado, como GroupBy o repartition, almacenan sus resultados en temas internos de Kafka. De forma predeterminada, Kafka Streams nombra estos temas internos en función del operador correspondiente. Si estos temas no existen, Kafka Streams crea temas internos de Kafka. Para crear los temas internos, Kafka Streams codifica la configuración segment.bytes y la establece en 50 MB. MSK aprovisionado con Express Brokers y MSK Serverless protege algunas configuraciones de temas, incluida segment.size, durante la creación del tema. Por lo tanto, una aplicación de Kafka Streams con transformaciones con estado no puede crear los temas internos con agentes de MSK Express o MSK Serverless.

Para ejecutar dichas aplicaciones de Kafka Streams en corredores de MSK Express o MSK Serverless, debe crear los temas internos usted mismo. Para ello, primero identifique y nombre a los operadores de Kafka Streams, que requieren temas. A continuación, cree los temas internos de Kafka correspondientes.

> **Note**
>
> - Se recomienda nombrar a los operadores manualmente en Kafka Streams, especialmente a los que dependen de temas internos. Para obtener información sobre cómo nombrar operadores, consulte Nombrar operadores en una aplicación DSL de Kafka Streams en la documentación de Kafka Streams.
> - El nombre del tema interno de una transformación con estado depende del nombre `application.id` de la aplicación de Kafka Streams y del nombre del operador con estado,. `application.id-statefuloperator_name`

Temas

- Creación de una aplicación de Kafka Streams con agentes de MSK Express o MSK Serverless

Creación de una aplicación de Kafka Streams con agentes de MSK Express o MSK Serverless

Si su aplicación de Kafka Streams `application.id` está configurada en`msk-streams-processing`, puede crear una aplicación de Kafka Streams con los corredores de MSK Express o MSK Serverless. Para ello, utilice el `count()` operador, que requiere un tema interno con el nombre. Por ejemplo, `msk-streams-processing-count-store`.

Para crear una aplicación Kafka Streams, haga lo siguiente:

Temas

- Identifique y asigne un nombre a los operadores
- Cree los temas internos
- (Opcional) Compruebe el nombre del tema
- Ejemplos de operadores de nomenclatura

Identifique y asigne un nombre a los operadores

1. Identifique los procesadores con estado mediante las transformaciones con estado de la documentación de Kafka Streams.

   Algunos ejemplos de procesadores con estado incluyen`count`, o. `aggregate` `join`
2. Identifique los procesadores que crean temas para la repartición.

   El siguiente ejemplo contiene una `count()` operación que necesita un estado.

   ```
   var stream =
       paragraphStream
           .groupByKey()
           .count()
           .toStream();
   ```
3. Para asignar un nombre al tema, añada un nombre para cada procesador con estado. Según el tipo de procesador, la denominación la realiza una clase de nomenclatura diferente. Por ejemplo, la `count()` operación es una operación de agregación. Por lo tanto, necesita la `Materialized` clase.

Para obtener información sobre las clases de nomenclatura de las operaciones con estado, consulte la conclusión en la documentación de Kafka Streams.

En el siguiente ejemplo, se establece el nombre del `count()` operador para que `count-store` utilice la `Materialized` clase.

```
var stream =
    paragraphStream
        .groupByKey()
        .count(Materialized.<String, Long, KeyValueStore<Bytes, byte[]>>as("count-
store")  // descriptive name for the store
            .withKeySerde(Serdes.String())
            .withValueSerde(Serdes.Long()))
        .toStream();
```

Cree los temas internos

Kafka transmite los prefijos de los nombres de los temas internos, siempre que los `application.id` defina el usuario`application.id`. Por ejemplo, `application.id-internal_topic_name`. Los temas internos son temas normales de Kafka y puede crearlos utilizando la información disponible en Creación de un tema de Apache Kafka o desde la API `AdminClient` de Kafka.

Según su caso de uso, puede utilizar las políticas de limpieza y retención predeterminadas de Kafka Streams o personalizar sus valores. Los defines en y. `cleanup.policy` `retention.ms`

En el siguiente ejemplo, se crean los temas con la `AdminClient` API y se establece `application.id` en**`msk-streams-processing`**.

```
try (AdminClient client = AdminClient.create(configs.kafkaProps())) {
    Collection<NewTopic> topics = new HashSet<>();
    topics.add(new NewTopic("msk-streams-processing-count-store", 3, (short) 3));
    client.createTopics(topics);
}
```

Una vez creados los temas en el clúster, la aplicación Kafka Streams puede usar el `msk-streams-processing-count-store` tema para la `count()` operación.

(Opcional) Compruebe el nombre del tema

Puede usar el descriptor de topografía para describir la topología de su arroyo y ver los nombres de los temas internos. El siguiente ejemplo muestra cómo ejecutar el descriptor de topología.

```
final StreamsBuilder builder = new StreamsBuilder();
Topology topology = builder.build();
System.out.println(topology.describe());
```

El siguiente resultado muestra la topología del flujo del ejemplo anterior.

```
Topology Description:
Topologies:
   Sub-topology: 0
    Source: KSTREAM-SOURCE-0000000000 (topics: [input_topic])
      --> KSTREAM-AGGREGATE-0000000001
    Processor: KSTREAM-AGGREGATE-0000000001 (stores: [count-store])
      --> KTABLE-TOSTREAM-0000000002
      <-- KSTREAM-SOURCE-0000000000
    Processor: KTABLE-TOSTREAM-0000000002 (stores: [])
      --> KSTREAM-SINK-0000000003
      <-- KSTREAM-AGGREGATE-0000000001
    Sink: KSTREAM-SINK-0000000003 (topic: output_topic)
      <-- KTABLE-TOSTREAM-0000000002
```

Para obtener información sobre cómo utilizar el descriptor de topología, consulte Nombrar operadores en una aplicación DSL de Kafka Streams en la documentación de Kafka Streams.

Ejemplos de operadores de nomenclatura

En esta sección se proporcionan algunos ejemplos de operadores de nomenclatura.

Ejemplo de operador de nomenclatura para groupByKey ()

```
groupByKey() -> groupByKey(Grouped.as("kafka-stream-groupby"))
```

Ejemplo de operador de nomenclatura para un recuento normal ()

```
normal count() -> .count(Materialized.<String, Long, KeyValueStore<Bytes,
 byte[]>>as("kafka-streams-window")  // descriptive name for the store
    .withKeySerde(Serdes.String())
```

```
    .withValueSerde(Serdes.Long()))
```

Ejemplo de operador de nomenclatura para windowed count ()

```
windowed count() -> .count(Materialized.<String, Long, WindowStore<Bytes,
 byte[]>>as("kafka-streams-window")  // descriptive name for the store
    .withKeySerde(Serdes.String())
    .withValueSerde(Serdes.Long()))
```

Ejemplo de operador de nomenclatura para windowed pressed ()

```
windowed suppressed() ->
Suppressed<Windowed> suppressed = Suppressed
        .untilWindowCloses(Suppressed.BufferConfig.unbounded())
        .withName("kafka-suppressed");
    .suppress(suppressed)
```

Planos de incrustación vectorial en tiempo real

Amazon MSK (Managed Streaming for Apache Kafka Kafka) es compatible con los planos de Amazon Managed Service for Apache Flink para generar incrustaciones vectoriales mediante Amazon Bedrock, lo que agiliza el proceso de creación de aplicaciones de IA en tiempo real basadas en datos contextuales. up-to-date El plan de MSF simplifica el proceso de incorporar los datos más recientes de sus canales de streaming de Amazon MSK a sus modelos de IA generativa, lo que elimina la necesidad de escribir código personalizado para integrar flujos de datos en tiempo real, bases de datos vectoriales y modelos de lenguaje de gran tamaño.

Puede configurar el plano de MSF para generar incrustaciones vectoriales de forma continua mediante los modelos de incrustación de Bedrock y, a continuación, indexar esas incrustaciones en Service para sus flujos de datos de Amazon MSK. OpenSearch Esto le permite combinar el contexto de los datos en tiempo real con los potentes modelos de lenguaje de gran tamaño de Bedrock para generar respuestas de IA precisas sin necesidad de escribir código personalizado. up-to-date También puedes optar por mejorar la eficiencia de la recuperación de datos mediante el soporte integrado para técnicas de fragmentación de datos desde una biblioteca de código abierto LangChain, que admite entradas de alta calidad para la incorporación de modelos. El plan gestiona la integración y el procesamiento de datos entre MSK, el modelo de incrustación elegido y el almacén de OpenSearch vectores, lo que le permite centrarse en crear sus aplicaciones de IA, en lugar de gestionar la integración subyacente.

Los planos de incrustación vectorial en tiempo real están disponibles en las siguientes regiones de: AWS

- Norte de Virginia: us-east-1
- Ohio: us-east-2
- Oregón: us-west-2
- Bombay: ap-south-1
- Seúl: ap-northeast-2
- Singapur: ap-southeast-1
- Sídney: ap-southeast-2
- Tokio: ap-northeast-1
- Centro de Canadá: ca-central-1
- Fráncfort: eu-central-1
- Irlanda: eu-west-1
- Londres: eu-west-2
- París: eu-west-3
- São Paulo: sa-east-1

Temas

- Registro y observabilidad
- Apuntes antes de habilitar los planos de incrustación vectorial en tiempo real
- Implemente un plan de vectorización de datos de streaming

Registro y observabilidad

Todos los registros y métricas para los planos de incrustación vectorial en tiempo real se pueden habilitar mediante registros. CloudWatch

Todas las métricas que están disponibles para una aplicación normal de MSF y Amazon Bedrock puede supervisar su aplicación y las métricas de Bedrock.

Hay dos métricas adicionales para monitorear el rendimiento de la generación de incrustaciones. Estas métricas forman parte del nombre de la EmbeddingGeneration operación en. CloudWatch

- BedrockTitanEmbeddingTokenCount: monitorea la cantidad de fichas presentes en una sola solicitud a Bedrock.
- BedrockEmbeddingGenerationLatencyMs: informa del tiempo que se tarda en enviar y recibir una respuesta de Bedrock para generar incrustaciones en milisegundos.

Para OpenSearch Service, puede utilizar las siguientes métricas de:

- OpenSearch Métricas de recopilación sin servidor: consulta Monitoring OpenSearch Serverless with Amazon CloudWatch en la Guía para desarrolladores de Amazon OpenSearch Service.
- OpenSearch métricas aprovisionadas: consulta Monitorización de métricas de OpenSearch clústeres con Amazon CloudWatch en la Guía para desarrolladores OpenSearch de Amazon Service.

Apuntes antes de habilitar los planos de incrustación vectorial en tiempo real

La aplicación de Managed Service para Apache Flink solo admitirá texto no estructurado o datos JSON en el flujo de entrada.

Se admiten dos modos de procesamiento de entradas:

- Cuando los datos de entrada son texto no estructurado, se incrusta todo el mensaje de texto. La base de datos vectorial contiene el texto original y la incrustación generada.
- Cuando los datos de entrada están en formato JSON, la aplicación permite configurar y especificar una o más claves dentro del valor del objeto JSON para utilizarlas en el proceso de incrustación. Si hay más de una clave, todas las claves se vectorizan y se indexan en la base de datos vectorial. La base de datos vectorial contendrá el mensaje original y la incrustación generada.

Generación de incrustaciones: la aplicación admite todos los modelos de incrustación de texto proporcionados exclusivamente por Bedrock.

Persistir en el almacén de bases de datos vectoriales: la aplicación utiliza un OpenSearch clúster existente (aprovisionado o sin servidor) en la cuenta del cliente como destino para los datos incrustados persistentes. Cuando utilice Opensearch Serverless para crear un índice vectorial, utilice siempre el nombre del campo vectorial. `embedded_data`

Al igual que en los planos de MSF, se espera que administres la infraestructura para ejecutar el código asociado al plan de incrustación vectorial en tiempo real.

Al igual que los Blueprints de MSF, una vez que se crea una aplicación de MSF, debe iniciarse exclusivamente en la AWS cuenta mediante la consola o la CLI. AWS no iniciará la aplicación de MSF por usted. Debe llamar a la StartApplication API (a través de la CLI o la consola) para que la aplicación se ejecute.

Movimiento de datos entre cuentas: la aplicación no permite mover datos entre el flujo de entrada y los destinos vectoriales que se encuentran en AWS cuentas diferentes.

Implemente un plan de vectorización de datos de streaming

En este tema se explica cómo implementar un blueprint de vectorización de datos de streaming.

Implemente un plan de vectorización de datos de streaming

1. Asegúrese de que los siguientes recursos estén configurados correctamente:
   - Clúster de MSK aprovisionado o sin servidor con uno o más temas que contienen datos.
2. Configuración de Bedrock: acceso al modelo de Bedrock deseado. Los modelos Bedrock compatibles actualmente son:
   - Amazon Titan Embeddings G1 - Text
   - Amazon Titan Text Embeddings V2
   - Amazon Titan Multimodal Embeddings G1
   - Cohere Embed inglés
   - Cohere Embed multilingüe
3. AWS OpenSearch colección:
   - Puede utilizar una colección de OpenSearch servicios aprovisionados o sin servidor.
   - La colección de OpenSearch servicios debe tener un índice como mínimo.
   - Si planea usar una colección OpenSearch sin servidor, asegúrese de crear una colección de búsqueda vectorial. Para obtener más información sobre cómo configurar un índice vectorial, consulta los requisitos previos de tu propio almacén de vectores para obtener una base de conocimientos. Para obtener más información sobre la vectorización, consulta la explicación de las capacidades de las bases de datos vectoriales de Amazon OpenSearch Service.

> **Note**
>
> Al crear un índice vectorial, debe usar el nombre del campo vectorial. `embedded_data`

- Si planea usar una colección OpenSearch aprovisionada, debe añadir el rol de aplicación MSF (que contiene la política de acceso de Opensearch) que creó el blueprint, como usuario maestro de su colección. OpenSearch Además, confirme que la política de acceso OpenSearch esté configurada para «Permitir» acciones. Esto es necesario para permitir el control de acceso al grano fino.
- Si lo desea, puede habilitar el acceso al OpenSearch panel de control para ver los resultados. Consulte esta sección para activar el control de acceso de grano fino.

4. Inicie sesión con un rol que permita CreateStack los permisos aws:.
5. Ve al panel de control de la consola de MSF y selecciona Crear aplicación de streaming.
6. En Elegir un método para configurar la aplicación de procesamiento de transmisiones, seleccione Usar un plano.
7. Seleccione el plano de la aplicación de IA en tiempo real en el menú desplegable de planos.
8. Proporcione las configuraciones deseadas. Consulte Creación de configuraciones de la página.
9. Seleccione Deploy Blueprint para iniciar una CloudFormation implementación.
10. Una vez que se complete la CloudFormation implementación, vaya a la aplicación Flink implementada. Compruebe las propiedades de tiempo de ejecución de la aplicación.
11. Puede elegir cambiar o añadir las propiedades de tiempo de ejecución a su aplicación. Consulte Configuración de propiedades en tiempo de ejecución para obtener detalles sobre cómo configurar estas propiedades.

> **Note**
>
> Nota:
> Si está utilizando OpenSearch aprovisionadas, asegúrese de haber activado el control de acceso detallado.
> Si el clúster aprovisionado es privado, agréguelo `https://` a la URL del punto de enlace de la OpenSearch VPC aprovisionada y cámbielo para que apunte `sink.os.endpoint` a este punto de enlace.

Si el clúster aprovisionado es público, asegúrate de que tu aplicación de MSF pueda acceder a Internet. Para obtener más información, consulte >>>>>> express-brokers-publication-merge type="documentation» url="managed- flink/latest/java/vpc - internet.html ">Acceso a Internet y a los servicios para una aplicación Managed Service for Apache Flink conectada a VPC.

12. Una vez esté satisfecho con todas las configuraciones, seleccioneRun. La aplicación empezará a ejecutarse.
13. Introduce los mensajes en tu clúster de MSK.
14. Navegue hasta el clúster de Opensearch y vaya al panel de OpenSearch control.
15. En el panel de control, selecciona Discover en el menú de la izquierda. Deberías ver los documentos persistentes junto con sus incrustaciones vectoriales.
16. Consulte Cómo utilizar las colecciones de búsqueda vectorial para ver cómo puede utilizar los vectores almacenados en el índice.

Creación de configuraciones de la página

En este tema, se describen las configuraciones de página de creación para consultarlas a la hora de especificar las configuraciones de los planos de aplicaciones de IA en tiempo real.

Nombre de la aplicación

Campo existente en MSF, asigne cualquier nombre a su aplicación.

Clúster de MSK

Seleccione el clúster de MSK que creó durante la configuración en la lista desplegable.

Temas

Añada el nombre de los temas que creó en la configuración.

Tipo de datos del flujo de entrada

Elija Cadena si va a proporcionar una entrada de cadena a la transmisión de MSK.

Elija JSON si la entrada de la transmisión de MSK es JSON. En las claves JSON integradas, escribe los nombres de los campos del JSON de entrada cuyo valor quieras enviar a Bedrock para generar incrustaciones.

Modelo de incrustación de Bedrock

Seleccione uno de la lista. Asegúrese de tener acceso al modelo que elija; de lo contrario, la pila podría fallar. Consulte Adición o eliminación del acceso a los modelos fundacionales de Amazon Bedrock.

OpenSearch clúster

Seleccione el clúster que creó en el menú desplegable.

OpenSearch nombre del índice vectorial

Seleccione el índice vectorial que creó en el paso anterior.

Cuota de Amazon MSK

Tienes Cuenta de AWS cuotas predeterminadas para Amazon MSK. A menos que se indique lo contrario, cada cuota por cuenta es específica de su región. Cuenta de AWS

Solicitar un aumento de cuota en Amazon MSK

Puede solicitar un aumento de cuota para cada región mediante la consola Service Quotas o un caso de soporte. AWS CLI Si no hay una cuota ajustable disponible en la consola de Service Quotas, utilice el caso AWS Support Center Console para crear un aumento de cuota de servicio.

Support podría aprobar, denegar o aprobar parcialmente sus solicitudes de aumento de cuota. Los aumentos no se conceden de inmediato y pueden tardar varios días en surtir efecto.

Para solicitar un aumento, visite la Consola de Service Quotas.

1. Abra la consola de Service Quotas en https://console.aws.amazon.com/servicequotas/.
2. Desde la barra de navegación, parte superior de la pantalla, seleccione una región.
3. En el panel de navegación izquierdo, elija Servicios de AWS.
4. En el cuadro Buscar servicios, escriba y, a continuación**msk**, elija Amazon Managed Streaming for Apache Kafka (MSK).
5. En Cuotas de servicio, elija el nombre de la cuota para la que desee solicitar un aumento. Por ejemplo, **Number of brokers per account**.
6. Elija Solicitud de aumento a nivel de cuenta.
7. En Aumentar el valor de la cuota, introduce un nuevo valor de cuota.
8. Seleccione Request (Solicitar).
9. (Opcional) Para ver las solicitudes pendientes o resueltas recientemente en la consola, selecciona Panel de control en el panel de navegación izquierdo. Para las solicitudes pendientes, seleccione el estado de la solicitud para abrir la recepción de solicitud. El estado inicial de una solicitud es Pendiente. Cuando el estado cambie a Cuota solicitada, verás el número de caso en Support. Elija el número de caso para abrir el ticket para su solicitud.

Para obtener más información, incluido cómo utilizar AWS CLI o SDKs solicitar un aumento de cuota, consulte Solicitar un aumento de cuota en la Guía del usuario de Service Quotas.

Cuota de bróker Amazon MSK Standard

Cuota de bróker estándar

Dimensión	Cuota	Notas
Corredores por cuenta	90	Para solicitar una cuota más alta, vaya a la consola de Service Quotas.
Corredores por grupo	30 para clústeres ZooKeeper basados, 60 para clústeres KRaft basados	Para solicitar una cuota más alta, vaya a la consola de Service Quotas.
Almacenamiento mínimo por agente	1 GiB	
Almacenamiento máximo por intermediario	16384 GiB	
Número máximo de conexiones TCP por intermediario (control de acceso IAM)	3 000	Para aumentar este límite, puede ajustar la propiedad de configuración `listener.name.client_iam.max.connections` o la propiedad de `listener.name.client_iam_public.max.connections` configuración mediante la `AlterConfig` API de Kafka o la `kafka-configs.sh` herramienta. Es importante tener en cuenta que aumentar cualquiera de las propiedades a un valor alto puede provocar la falta de disponibilidad.

Dimensión	Cuota	Notas
Velocidad máxima de conexiones TCP por intermediario (IAM)	100 por segundo (tamaños de instancia M5 y M7g) 4 por segundo (tamaño de instancia t3)	Para gestionar los reintentos en caso de conexiones fallidas, puede configurar el parámetro de configuración `reconnect.backoff.ms` en el lado del cliente. Por ejemplo, si quiere que un cliente vuelva a intentar conectarse después de 1 segundo, configúrelo en. `reconnect.backoff.ms 1000` Para obtener más información, consulte reconnect.backoff.ms en la documentación de Apache Kafka.
Número máximo de conexiones TCP por intermediario (no IAM)	N/A	MSK no impone límites de conexión para la autenticación que no sea de IAM. Debe supervisar otras métricas, como el uso de la CPU y la memoria, para asegurarse de no sobrecargar el clúster debido a un exceso de conexiones.

Dimensión	Cuota	Notas
Configuraciones por cuenta de	100	Para solicitar una cuota más alta, vaya a la consola de Service Quotas. Para actualizar la configuración o la versión de Apache Kafka de un clúster de MSK, primero asegúrese de que el número de particiones por agente esté por debajo de los límites descritos en Ajuste el tamaño correcto del clúster: número de particiones por agente.
Revisiones de configuración por cuenta	50	

Cuota de bróker de Amazon MSK Express

Cuotas exprés para corredores

Dimensión	Cuota	Notas
Corredores por cuenta	90	Para solicitar una cuota más alta, vaya a la consola de Service Quotas.
Corredores por grupo	30	Para solicitar una cuota más alta, vaya a la consola de Service Quotas.
Almacenamiento máximo	Sin límite	

Dimensión	Cuota	Notas
Número máximo de conexiones TCP por intermediario (control de acceso IAM)	3 000	Para aumentar el límite de conexión, ajuste una de las siguientes propiedades de configuración mediante la AlterConfig API de Kafka o la herramienta kafka-configs.sh: • `listener.name.client_iam.max.connections` • `listener.name.client_iam_public.max.connections` Si se establecen estas propiedades en un valor alto, es posible que el clúster no esté disponible.

Dimensión	Cuota	Notas
Velocidad máxima de conexiones TCP por intermediario (IAM)	100 por segundo	Para gestionar los reintentos en caso de conexiones fallidas, puede configurar el parámetro de configuración `reconnect.backoff.ms` en el lado del cliente. Por ejemplo, si desea que un cliente vuelva a intentar conectarse después de 1 segundo, `reconnect.backoff.ms` configúrelo en. `1000` Para obtener más información, consulte reconnect.backoff.ms en la documentación de Apache Kafka.
Número máximo de conexiones TCP por intermediario (no IAM)	N/A	MSK no impone límites de conexión para la autenticación que no sea de IAM. Sin embargo, debe supervisar otras métricas, como el uso de la CPU y la memoria, para asegurarse de no sobrecargar el clúster debido a un exceso de conexiones.

Dimensión	Cuota	Notas
Configuraciones por cuenta de	100	Para solicitar una cuota más alta, vaya a la consola de Service Quotas. Para actualizar la configuración o la versión de Apache Kafka de un clúster de MSK, primero asegúrese de que el número de particiones por agente esté por debajo de los límites descritos en Ajuste el tamaño correcto de su clúster: número de particiones por broker Express.
Revisiones de configuración por cuenta	50	
Entrada máxima por bróker	Recomendado: 15,6 - 500,0 MBps	Basado en el tamaño de la instancia.
Salida máxima por corredor	Recomendado: 31,2 - 1000,0 MBps	Basado en el tamaño de la instancia.

Limita el rendimiento de Express Broker según el tamaño del bróker

En la siguiente tabla se enumeran los límites de rendimiento máximo y recomendado en relación con la entrada y la salida para corredores de distintos tamaños. En esta tabla, el rendimiento recomendado se representa como rendimiento sostenido, que es el umbral hasta el que sus aplicaciones no experimentarán ninguna degradación del rendimiento. Si opera más allá de estos límites en cualquiera de las dimensiones, es posible que obtenga un mayor rendimiento, pero también que experimente una degradación del rendimiento. La cuota máxima es el umbral en el que el clúster limitará el tráfico. read/write Sus aplicaciones no podrán funcionar más allá de este umbral.

Tamaño de instancia	Rendimiento sostenido (MBps) para la entrada	Cuota máxima (MBps) de entrada	Rendimiento sostenido (MBps) para la salida	Cuota máxima (MBps) de salida
express.m7g.large	15.6	23.4	31,2	58,5
express.m7g.xlarge	31.2	46,8	62,5	117
express.m7g.2xlarge	62,5	93,7	125	234,2
express.m7g.4xlarge	124.9	187,5	249,8	468,7
express.m7g.8xlarge	250	375	500	937,5
express.m7g. 12 x grande	375	562,5	750	1406.2
express.m7g. 16 x grande	500	750	1 000	1875

Cuotas del Replicador MSK

- Un máximo de 15 replicadores de MSK por cuenta.
- El Replicador MSK solo replica hasta 750 temas en orden. Si necesita replicar más temas, le recomendamos que cree un Replicador independiente. Si necesita ayuda para más de 750 temas por Replicador, vaya a la consola de Service Quotas. Puede controlar la cantidad de temas que se replican mediante la métrica «TopicCount».
- Un rendimiento de entrada máximo de 1 GB por segundo por Replicador MSK. Solicite una cuota más alta desde la consola de Service Quotas.
- Tamaño de registro del Replicador MSK: un tamaño de registro máximo de 10 MB (message.max.bytes). Solicite una cuota más alta desde la consola de Service Quotas.

Cuota de MSK sin servidor

Las cuotas especificadas en la siguiente tabla son por clúster, a menos que se indique lo contrario.

> **Note**
>
> Si tiene algún problema con los límites de las cuotas de servicio, cree un caso de soporte e incluya su caso de uso y el límite solicitado.

Dimensión	Cuota	Resultado de infracción de cuota
Rendimiento máximo de entrada	200 MBps	Ralentización con la duración del acelerador en respuesta
Rendimiento máximo de salida	400 MBps	Ralentización con la duración del acelerador en respuesta
Duración máxima de retención	Sin límite	N/A
Número máximo de conexiones de cliente	3 000	Cierre de conexión
Número máximo de intentos de conexión	100 por segundo	Cierre de conexión
Tamaño máximo de mensaje	8 MiB	La solicitud falla con ErrorCode: INVALID_REQUEST
Tasa máxima de solicitud	15 000 por segundo	Ralentización con la duración del acelerador en respuesta
Tasa máxima de solicitudes de gestión de temas APIs	2 por segundo	Ralentización con la duración del acelerador en respuesta

Dimensión	Cuota	Resultado de infracción de cuota
Número máximo de bytes de recuperación por solicitud	55 MB	La solicitud falla con ErrorCode: INVALID_REQUEST
Número máximo de grupos de consumidores	500	JoinGroup la solicitud falla
Número máximo de particiones (principales)	2400 para temas no compactados. 120 para temas compactados. Para solicitar un ajuste de la cuota de servicio, cree un caso de soporte e incluya su caso de uso y el límite solicitado.	La solicitud falla con ErrorCode: INVALID_REQUEST
Velocidad máxima de creación y eliminación de particiones	250 en 5 minutos	La solicitud falla con: THROUGHPUT_QUOTA_EXCEEDED ErrorCode
Rendimiento máximo de entrada por partición	5 MBps	Ralentización con la duración del acelerador en respuesta
Rendimiento máximo de salida por partición	10 MBps	Ralentización con la duración del acelerador en respuesta
Tamaño máximo de partición (para temas compactados)	250 GB	La solicitud falla con ErrorCode: THROUGHPUT_QUOTA_EXCEEDED
Número máximo de clientes por clúster sin servidor VPCs	5	

Dimensión	Cuota	Resultado de infracción de cuota
Número máximo de clústeres sin servidor por cuenta	10. Para solicitar un ajuste de la cuota de servicio, cree un caso de soporte e incluya su caso de uso y el límite solicitado.	

Cuota de MSK Connect

- Hasta 100 complementos personalizados.
- Configuraciones de hasta 100 procesos de trabajo.
- Hasta 60 procesos de trabajo conectados. Si un conector está configurado para que tenga una capacidad que se escale automáticamente, el número máximo de procesos de trabajo que puede tener el conector es el número que MSK Connect utiliza para calcular la cuota de la cuenta.
- Hasta 10 procesos de trabajo por conector.

Para solicitar una cuota más alta para MSK Connect, vaya a la consola de Service Quotas.

Historial de documento de la Guía para desarrolladores de Amazon MSK

En la siguiente tabla se describen los cambios importantes que se han realizado en la Guía para desarrolladores de Amazon MSK.

Actualización de la documentación más reciente: 25 de junio de 2024

Cambio	Descripción	Fecha
Se agregó la característica Express Broker. Se han reorganizado los temas de la Guía para desarrolladores.	MSK es compatible con los corredores Standard y los nuevos de Express.	2024-11-6
Se agregó la característica de actualización inmediata de Graviton.	Puede actualizar el tamaño del agente de clúster de M5 o T3 a M7g, o de M7g a M5.	2024-6-25
Se anunció la fecha de finalización del soporte de la versión 3.4.0.	La fecha de finalización del soporte de la versión 3.4.0 de Apache Kafka es el 17 de junio de 2025.	2024-6-24
Se agregó la característica de eliminación de agentes.	Puede reducir la capacidad de almacenamiento y computación del clúster aprovisionado al eliminar grupos de agentes, sin que ello afecte la disponibilidad, ponga en riesgo la durabilidad de los datos ni interrumpa las aplicaciones de flujo de datos.	2024-5-16
`WriteDataIdempotently` añadido a AWSMSKReplicator ExecutionRole	WriteDataIdempotently Se agregó el permiso a la AWSMSKReplicator Execution	16-05-2022

Cambio	Descripción	Fecha
	Role política para la compatibilidad con la replicación de datos entre los clústeres de MSK.	
Se lanzaron los agentes de M7g de Graviton en Brasil y Bahréin.	Amazon MSK ahora es compatible con la disponibilidad de agentes de M7g en las regiones de Sudamérica (sa-east-1, São Paulo) y Medio Oriente (me-south-1, Baréin), con procesadores Graviton (AWS procesadores personalizados basados en Arm, desarrollados por Amazon Web Services).	7 de febrero de 2024
Lanzamiento de los agentes de M7g de Graviton en la región de China	Amazon MSK ahora es compatible con la disponibilidad de agentes de M7g en la región de China, con procesadores AWS Graviton (procesadores personalizados basados en Arm, desarrollados por Amazon Web Services).	2024-01-11
Política de soporte de las versiones de Amazon MSK Kafka	Se agregó una explicación de la política de soporte de las versiones de Kafka compatible con Amazon MSK. Para obtener más información, consulte las versiones de Apache Kafka.	2023-12-08

Cambio	Descripción	Fecha
Nueva política de roles de ejecución de servicios para la compatibilidad con el Replicador Amazon MSK.	Amazon MSK agregó una nueva política de `AWSMSKReplicatorExecutionRole` para la compatibilidad con el Replicador Amazon MSK. Para obtener más información, consulte Políticas administradas de AWS : `AWSMSKReplicatorExecutionRole`.	2023-12-06
Compatibilidad con M7g de Graviton	Amazon MSK ahora es compatible con los agentes de M7g con procesadores AWS Graviton (procesadores personalizados basados en Arm, desarrollados por Amazon Web Services).	2023-11-27
Replicador Amazon MSK	El Replicador Amazon MSK es una nueva característica que puede utilizar para replicar datos entre clústeres de Amazon MSK. El Replicador Amazon MSK incluye una actualización de la política de Amazon MSKFull Access. Para obtener más información, consulte Políticas administradas de AWS : `AmazonMSKFullAccess`.	2023-09-28

Cambio	Descripción	Fecha
Se ha actualizado para las prácticas recomendadas de IAM.	Se ha actualizado la guía para implementar las prácticas recomendadas de IAM. Para obtener más información, consulta prácticas recomendadas de seguridad en IAM.	2023-03-08
Actualizaciones de funciones vinculadas al servicio para permitir la conectividad privada de varias VPC	Amazon MSK ahora incluye actualizaciones de roles AWSService RoleForKafka vinculados al servicio para administrar las interfaces de red y los puntos de conexión de VPC en su cuenta, lo que permite que los agentes de clúster sean accesibles para los clientes de su VPC. Amazon MSK usa permisos para `DescribeVpcEndpoints` , `ModifyVpcEndpoint` y `DeleteVpcEndpoints` . Para obtener más información, consulte Roles vinculados a servicios para Amazon MSK.	2023-03-08
Soporte de Apache Kafka 2.7.2	Amazon MSK ahora es compatible con la versión 2.7.2 de Apache Kafka. Para obtener más información, consulte Versiones compatibles de Apache Kafka.	2021-12-21

Cambio	Descripción	Fecha
Soporte de Apache Kafka 2.6.3	Amazon MSK ahora es compatible con la versión 2.6.3 de Apache Kafka. Para obtener más información, consulte Versiones compatibles de Apache Kafka.	2021-12-21
Versión preliminar de MSK sin servidor.	MSK sin servidor es una nueva característica que puede utilizar para crear clústeres sin servidor. Para obtener más información, consulte MSK sin servidor.	2021-11-29
Soporte de Apache Kafka 2.8.1	Amazon MSK ahora es compatible con la versión 2.8.1 de Apache Kafka. Para obtener más información, consulte Versiones compatibles de Apache Kafka.	2021-09-30
MSK Connect	MSK Connect es una nueva característica que puede usar para crear y administrar conectores de Apache Kafka. Para obtener más información, consulte Descripción de MSK Connect.	2021-09-16
Soporte de Apache Kafka 2.7.1	Amazon MSK ahora es compatible con la versión 2.7.1 de Apache Kafka. Para obtener más información, consulte Versiones compatibles de Apache Kafka.	2021-05-25

Cambio	Descripción	Fecha
Soporte de Apache Kafka 2.8.0	Amazon MSK ahora es compatible con la versión 2.8.0 de Apache Kafka. Para obtener más información, consulte Versiones compatibles de Apache Kafka.	2021-04-28
Soporte de Apache Kafka 2.6.2	Amazon MSK ahora es compatible con la versión 2.6.2 de Apache Kafka. Para obtener más información, consulte Versiones compatibles de Apache Kafka.	2021-04-28
Soporte para actualizar el tipo de agente	Ahora puede cambiar el tipo de agente de un clúster existente. Para obtener más información, consulte Actualización del tamaño del agente del clúster de Amazon MSK.	2021-01-21
Soporte de Apache Kafka 2.6.1	Amazon MSK ahora es compatible con la versión 2.6.1 de Apache Kafka. Para obtener más información, consulte Versiones compatibles de Apache Kafka.	2021-01-2021
Soporte de Apache Kafka 2.7.0	Amazon MSK ahora es compatible con la versión 2.7.0 de Apache Kafka. Para obtener más información, consulte Versiones compatibles de Apache Kafka.	2020-12-29

Cambio	Descripción	Fecha
No hay clústeres nuevos con la versión 1.1.1 de Apache Kafka	Ya no puede crear un nuevo clúster de Amazon MSK con la versión 1.1.1 de Apache Kafka. Sin embargo, si tiene clústeres de MSK existente s que ejecutan la versión 1.1.1 de Apache Kafka, puede seguir utilizando todas las características compatibles actualmente en esos clústeres existentes. Para obtener más información, consulte Versiones de Apache Kafka.	2020-11-24
Métricas de retraso entre los consumidores	Amazon MSK proporciona ahora métricas que se pueden utilizar para monitorizar el retraso de consumo. Para obtener más información, consulte Supervise un clúster aprovisionado de Amazon MSK.	2020-11-23
Soporte de Cruise Control	Amazon MSK ahora LinkedIn es compatible con Cruise Control. Para obtener más información, consulte Utilice LinkedIn el control de crucero para Apache Kafka con Amazon MSK.	2020-11-17

Cambio	Descripción	Fecha
Soporte de Apache Kafka 2.6.0	Amazon MSK ahora es compatible con la versión 2.6.0 de Apache Kafka. Para obtener más información, consulte Versiones compatibles de Apache Kafka.	2020-10-21
Soporte de Apache Kafka 2.5.1	Amazon MSK ahora es compatible con la versión 2.5.1 de Apache Kafka. Con la versión 2.5.1 de Apache Kafka, Amazon MSK admite el cifrado en tránsito entre los clientes y los puntos de conexión. ZooKeeper Para obtener más información, consulte Versiones compatibles de Apache Kafka.	2020-09-30
Expansión automática de aplicaciones	Puede configurar Amazon Managed Streaming para Apache Kafka para ampliar automáticamente el almacenamiento de su clúster en respuesta al aumento del uso. Para obtener más información, consulte Escalado automático para clústeres.	2020-09-30

Cambio	Descripción	Fecha
Soporte de seguridad de nombre de usuario y contraseña	Amazon MSK ahora admite el inicio de sesión en clústeres con un nombre de usuario y una contraseña. Amazon MSK almacena las credenciales en AWS Secrets Manager. Para obtener más información, consulte Autenticación SASL/SCRAM.	2020-09-17
Soporte para actualizar la versión de Apache Kafka de un clúster de Amazon MSK	Ahora puede actualizar la versión de Apache Kafka de un clúster de MSK existente.	2020-05-28
Soporte para nodos de agente T3.Small	Amazon MSK ahora admite la creación de clústeres con agentes de Amazon EC2 tipo T3.small.	08/04/2020
Soporte de Apache Kafka 2.4.1	Amazon MSK ahora es compatible con la versión 2.4.1 de Apache Kafka.	02/04/2020
Compatibilidad con la transmisión de registros de agente	Amazon MSK ahora puede transmitir los registros de agente a CloudWatch Registros de Logs, Amazon S3 y Amazon Data Firehose. Firehose puede, a su vez, entregar estos registros a los destinos compatibles, como OpenSearch Service.	25/02/2020
Soporte de Apache Kafka 2.3.1	Amazon MSK ahora es compatible con la versión 2.3.1 de Apache Kafka.	19/12/2019

Cambio	Descripción	Fecha
Monitoreo abierto	Amazon MSK ahora es compatible con el monitoreo abierto con Prometheus.	04/12/2019
Soporte de Apache Kafka 2.2.1	Amazon MSK ahora es compatible con la versión 2.2.1 de Apache Kafka.	31/07/2019
Disponibilidad general	Entre las nuevas caracterí sticas se incluyen la asistenci a para el etiquetado, autentica ción, cifrado de TLS, configura ciones y la capacidad de actualizar almacenamiento de agentes.	30/05/2019
Soporte de Apache Kafka 2.1.0	Amazon MSK ahora es compatible con la versión 2.1.0 de Apache Kafka.	05/02/2019

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.