



Guía para desarrolladores

Transmisión administrada de Amazon para Apache Kafka



Transmisión administrada de Amazon para Apache Kafka: Guía para desarrolladores

Copyright © 2026 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

Welcome	1
¿Qué es Amazon MSK?	1
Configuración	3
Inscríbase en AWS	3
Descargar bibliotecas y herramientas	3
MSK aprovisionado	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	18
Ver las métricas de	19
Eliminación de los recursos del tutorial	19
Funcionamiento	20
Administración del clúster de MSK aprovisionado	20
Creación de un clúster	21
Enumerar clústeres	32
Conexión a un clúster de MSK aprovisionado	33
Obtención de los agentes de arranque	56
Supervisión de un clúster	58
Actualización de la seguridad del clúster	106
Ampliación de un clúster	109
Eliminación de un agente	112
Actualización del tamaño del agente del clúster	117
Uso de Cruise Control	121
Actualización de la configuración de un clúster	126
Configure el tipo de red de doble pila	130
Reinicio de un agente para un clúster de Amazon MSK	134
Etiquetado de un clúster	136
Migración a un clúster de MSK	138
Eliminar un clúster	139
Características y conceptos clave	140
Tipos de agente	141

Tamaños de agentes	145
Administrar el almacenamiento	147
Configuraciones de agentes	168
Reequilibrio inteligente	244
Aplicación de parches en clústeres aprovisionados	249
El agente no está en línea y el cliente realiza una conmutación por error	251
Seguridad	254
Registro de Amazon MSK	328
Administración de metadatos	336
Tema: Operaciones	340
Recursos	354
Versiones de Apache Kafka	354
Solución de problemas del clúster de Amazon MSK	369
Prácticas recomendadas	380
Prácticas recomendadas para agentes Standard	380
Prácticas recomendadas para agentes Express	389
Prácticas recomendadas para clientes de Apache Kafka	394
MSK sin servidor	402
Uso de clústeres sin servidor de MSK	403
Creación de un clúster	403
Creación de un rol de IAM	405
Creación de un equipo cliente	407
Crear un tema	409
Producción y consumo de datos	411
Eliminación de recursos	412
Propiedades de configuración	413
Configure el tipo de red de doble pila	414
Requisitos previos	415
Permisos de IAM	415
Utilice un tipo de red de doble pila	415
Consideraciones	417
Supervisión	417
MSK Connect	420
Beneficios de Amazon MSK Connect	420
Introducción	422
Configuración de los recursos necesarios para MSK Connect	422

Creación de un complemento personalizado	427
Creación de un equipo cliente y un tema de Apache Kafka	428
Crea conector	431
Envío de datos al clúster de MSK	432
Descripción de los conectores	433
Descripción de la capacidad de los conectores	434
Conozca el número máximo de tareas de escalado automático	435
Configure el tipo de red de doble pila	437
Creación de un conector de	438
Actualización de un conector	440
Conexión desde conectores	441
Creación de complementos personalizados	441
Descripción de los procesos de trabajo de MSK Connect	443
Configuración predeterminada del proceso de trabajo	444
Propiedades de configuración de proceso de trabajo compatibles	444
Creación de una configuración personalizada	446
Administración de los desplazamientos de los conectores	447
Proveedores de configuración	451
Consideraciones	451
Creación de un complemento personalizado y carga a S3	452
Configuración de los parámetros y permisos para los distintos proveedores	454
Creación de una configuración personalizada de proceso de trabajo	458
Creación del conector	459
Roles y políticas de IAM	460
Descripción del rol de ejecución de servicios	460
Ejemplo de política de	463
Prevención de los problemas del suplente confuso entre servicios	465
AWS políticas gestionadas	467
Uso de roles vinculados a servicios	471
Habilitar el acceso a Internet	473
Configuración de una puerta de enlace NAT	473
Descripción de los nombres de host de DNS privados	475
Configuración de una opción de DHCP de la VPC	476
Configuración de atributos de DNS	477
Solución de errores en la creación de conectores	477
Seguridad	478

Registro	478
Impedir que los secretos aparezcan en los registros de los conectores	480
Supervisión de MSK Connect	480
Ejemplos	484
Configuración de un conector de recepción de Amazon S3	484
Configure el conector del colector EventBridge Kafka	486
Uso del conector de origen Debezium	492
Migración a Amazon MSK Connect	503
Descripción de los temas internos que utiliza Kafka Connect	504
Administración de estados	505
Migración de los conectores de origen	505
Migración de los conectores de recepción	506
Resolución de problemas	508
Replicador MSK	509
Introducción	510
Configuraciones admitidas	511
Requisitos previos	512
Prepare los clústeres de origen y destino	513
Cree su primer replicador	516
Operaciones de replicador	518
Cree un replicador	518
Edite un replicador	529
Eliminar un replicador	530
Supervisar la replicación	531
Patrones de replicación	531
Replicación en la misma región	532
Replicación entre regiones	532
Replicación activa-pasiva	533
Replicación activa-activa	533
Conmutación por error a la región secundaria	535
Conmutación por error planificada	535
Conmutación por error no planificada	536
Recaída por error	537
Migración	539
Migre entre clústeres de Amazon MSK	539
Migre de Kafka autogestionado	540

Migre de 2 MirrorMaker	549
Cross-account migración	549
Conceptos clave	549
Cómo funciona la replicación	550
Denominación de temas	551
Replicación de metadatos y ACL	552
Sincronización compensada por grupos de consumidores	554
Requisitos de red	558
Seguridad	559
Función de ejecución de servicios (SER)	559
Cifrado	562
Permisos de IAM necesarios	562
Referencia de permisos de IAM	567
Precios	568
Observabilidad	569
Referencia de métricas	569
Supervise la replicación desde clústeres Kafka autogestionados	589
Registros de MSK Replicator	590
Prácticas recomendadas	592
Dimensionamiento y particiones	592
Administrar el rendimiento con cuotas de Kafka	593
Periodo de retención	594
La configuración del clúster cambia tras la creación del replicador	595
Consideraciones para aplicaciones multirregionales	595
Resolución de problemas	596
El estado del replicador pasa de CREATIVO a FALLIDO	596
El replicador parece atascado en el estado DE CREACIÓN	596
Replicator no replica datos o solo replica datos parciales	596
Los desplazamientos de mensajes en el clúster de destino son diferentes a las del clúster de origen	597
Replicator no sincroniza las compensaciones de grupos de consumidores	597
La latencia de replicación es alta o sigue aumentando	598
Solución de problemas mediante la métrica ReplicatorFailure	599
Solucione los problemas de replicación desde clústeres Kafka autogestionados	603
Integraciones de MSK	606
Conector de Athena para Amazon MSK	606

Integración de Redshift en Amazon MSK	606
Integración de Firehose en Amazon MSK	607
Integración de Lambda con Amazon MSK	607
Tubos de acceso EventBridge	607
Uso de Kafka Streams con agentes Express y Amazon MSK sin servidor	609
Creación de una aplicación de Kafka Streams	610
Esquemas de incrustación vectorial en tiempo real	613
Registro y observabilidad	615
Notas antes de habilitar esquemas de incrustación vectorial en tiempo real	615
Implementación de un esquema de vectorización de datos en streaming	616
Cuota	620
Solicitud de un aumento de cuota en Amazon MSK	620
Cuota estándar de agentes	621
Cuota de agentes Express	623
Límites de limitación controlada del rendimiento de agentes Express según el tamaño del agente	626
Cuota de particiones de agentes Express	627
Cuotas del Replicador MSK	628
Cuota para clústeres sin servidor	628
Cuota de MSK Connect	630
Historial de revisión	631
.....	dcl

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

Bienvenido a la Guía para desarrolladores de Amazon Managed Streaming para 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 de MSK aprovisionado según el tutorial de [Introducción a Amazon MSK](#).
- Profundice en la funcionalidad de MSK aprovisionado en [¿Qué es MSK aprovisionado?](#).
- Ejecute Apache Kafka sin tener que administrar ni escalar la capacidad del clúster con [MSK sin servidor](#).
- Utilice [MSK Connect](#) para transmitir datos hacia el clúster de Apache Kafka y desde este.
- Utilice [MSK Replicator](#) para replicar datos de forma confiable entre clústeres de MSK aprovisionado en diferentes Regiones de AWS o en la misma.

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 agentes:** al crear un clúster de Amazon MSK, especifique cuántos nodos de agentes 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 aprovisionado ofrece dos tipos de agentes: [Agentes Standard de Amazon MSK](#) y [Agentes Express de Amazon MSK](#). En [MSK sin servidor](#), MSK administra los nodos de agentes utilizados para gestionar el tráfico, por lo que solo debe aprovisionar los recursos del servidor de Kafka en el 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 KRaft Los 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 Consola de administración de AWS, 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 AWS se 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 aprovisionado?

Los clústeres de Amazon MSK aprovisionado ofrecen una amplia gama de características y capacidades que ayudan a optimizar el rendimiento del clúster y a satisfacer las necesidades de streaming. Los temas que se muestran a continuación describen la funcionalidad en detalle.

MSK aprovisionado es una opción de implementación de clústeres de MSK que permite configurar y escalar manualmente los clústeres de Apache Kafka. Esta opción proporciona distintos niveles de control sobre la infraestructura que sustenta el entorno de Apache Kafka. Con MSK aprovisionado, puede elegir los tipos de instancia, los volúmenes de almacenamiento (para agentes Standard) y la cantidad de nodos de agente que conforman los clústeres de Kafka. También puede escalar el clúster al agregar o eliminar agentes conforme evolucionan las necesidades de procesamiento de datos. Esta flexibilidad permite optimizar el clúster según los requisitos específicos de la carga de trabajo, 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 aprovisionado ofrece seguridad, supervisión y beneficios operativos 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 aprovisionado ofrece dos tipos principales de agentes: Standard y Express.

Para obtener información sobre la API de MSK aprovisionado, 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: Creación de un clúster de MSK aprovisionado](#)
- [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: Creación de un clúster de MSK aprovisionado

En este paso de [Introducción al uso de Amazon MSK](#), crea un clúster de Amazon MSK aprovisionado. Para crear este clúster, utilice la opción Consola de administración de AWS de creación rápida.

Para crear un clúster de Amazon MSK mediante el Consola de administración de AWS

1. ¿Iniciar sesión en la Consola de administración de AWS 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. En Propiedades generales del clúster, realice lo siguiente:
 - a. En Tipo de clúster, seleccione Aprovisionado.
 - b. Seleccione una versión de Apache Kafka que se ejecutará en los agentes. Seleccione Ver compatibilidad de versiones para consultar la tabla comparativa.
 - c. En Tipo de agente, elija Standard o Express.
 - d. Seleccione un Tamaño de agente.
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 [Descripción de los estados de los clústeres de MSK aprovisionados](#).

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 Editor de políticas, elija JSON y, a continuación, reemplace el JSON en la ventana del editor por el siguiente JSON.

En el ejemplo siguiente, sustituya:

- *region* con el código del lugar donde creaste tu clúster. Región de AWS
- Ejemplo de ID de cuenta *123456789012*, con tu Cuenta de AWS ID.
- *MSKTutorialCluster* *MSKTutorialCluster/7d7131e1-25c5-4e9a-9ac5-ea85bee4da11-14*, con el nombre del clúster y su ID.

JSON

```
{
```

```

    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "kafka-cluster:Connect",
          "kafka-cluster:AlterCluster",
          "kafka-cluster:DescribeCluster"
        ],
        "Resource": [
          "arn:aws:kafka:us-east-1:123456789012:cluster/MSKTutorialCluster/7d7131e1-25c5-4e9a-9ac5-ea85bee4da11-14"
        ]
      },
      {
        "Effect": "Allow",
        "Action": [
          "kafka-cluster:*Topic*",
          "kafka-cluster:WriteData",
          "kafka-cluster:ReadData"
        ],
        "Resource": [
          "arn:aws:kafka:us-east-1:123456789012:topic/MSKTutorialCluster/*"
        ]
      },
      {
        "Effect": "Allow",
        "Action": [
          "kafka-cluster:AlterGroup",
          "kafka-cluster:DescribeGroup"
        ],
        "Resource": [
          "arn:aws:kafka:us-east-1:123456789012:group/MSKTutorialCluster/*"
        ]
      }
    ]
  }
}

```

Para obtener instrucciones sobre cómo escribir políticas seguras, consulte [the section called “Control de acceso de IAM”](#).

5. Elija Siguiente.

6. En la página Revisar y crear, haga lo siguiente:
 - a. En Nombre de la política, introduzca un nombre descriptivo, como **msk-tutorial-policy**.
 - b. En Permisos definidos en esta política, revise y and/or edita los permisos definidos en tu 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, agregue una etiqueta a la política con el par clave-valor de **Environment** y **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, seleccione Roles y luego seleccione Crear rol.
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 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, introduzca el nombre de la política que creó previamente para este tutorial. A continuación, marque 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 [Creación de roles y asociación de 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 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 tu Cuenta de AWS perfil y no se pueden hacer 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.

- (Opcional) En Descripción, ingrese una descripción para el rol.
- (Opcional) Para editar los casos de uso y los permisos del rol, en las secciones Paso 1: Seleccionar entidades de confianza o Paso 2: Agregar permisos, elija Editar.
- (Opcional) Para ayudar a identificar, organizar o buscar el rol, elija Agregar nueva etiqueta para agregar etiquetas como pares clave-valor. Por ejemplo, agregue una etiqueta al rol con el par clave-valor de **ProductManager** y **John**.

Para obtener más información sobre el uso de etiquetas, consulte [Etiquetas para obtener 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. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
2. En el panel de la consola de Amazon EC2, elija Iniciar instancia.
3. En Nombre y etiquetas, en Nombre, introduzca un nombre descriptivo para la máquina cliente, de modo que pueda identificarla y realizar un seguimiento de esta con facilidad. Por ejemplo, **MSKTutorialClient**.
4. En Aplicación e imágenes del sistema operativo (imagen de máquina de Amazon), en Imagen de máquina de Amazon (AMI), seleccione AMI de Amazon Linux 2 (HVM) – Kernel 5.10, tipo de volumen SSD.
5. En Tipo de instancia, mantenga la selección predeterminada t2.micro.
6. En Par de claves (inicio de sesión), seleccione un par de claves existente o cree uno nuevo. Si no necesita un par de claves para conectarse a la instancia, puede seleccionar Continuar sin un par de claves (no recomendado).

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

- a. Seleccione Crear un nuevo par de claves.
- b. En Key pair name (Nombre del par de claves), ingrese **MSKKeyPair**.
- c. En Tipo de par de claves y Formato del 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 Detalles avanzados. A continuación, realice lo siguiente:
 - En Perfil de instancia de IAM, seleccione un rol de IAM que desee que la máquina cliente asuma.

Si no dispone de un rol de IAM, realice lo siguiente:

- i. Seleccione Crear un nuevo perfil de IAM.
- ii. Complete los pasos descritos en el [Paso 2: Creación de 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](#), puede crear un tema con uno de estos dos enfoques: usar AWS herramientas nativas con la CreateTopic API o usar AdminClient herramientas de Apache Kafka en un equipo cliente.

Warning

Cuando utilice AWS herramientas con la CreateTopic API, compruebe que el clúster cumpla los requisitos. Para obtener más información, consulte [el tema Requisitos de uso APIs](#).

Warning

Al utilizar este AdminClient enfoque, 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 una versión anterior del cliente carezca de determinadas características y correcciones críticas de errores.

Temas

- [Crear un tema mediante herramientas AWS](#)
- [Determinación de la versión del clúster de MSK](#)
- [Creación de un tema en la máquina cliente](#)

Crear un tema mediante herramientas AWS

Puede crear temas en su clúster de MSK mediante AWS herramientas como la AWS CLI o la consola AWS de administración. AWS SDKs Este enfoque proporciona una forma simplificada de gestionar los temas sin necesidad de acceso directo a las herramientas de cliente de Kafka.

Para obtener información detallada sobre la creación de temas con las AWS herramientas, consulte la guía para [desarrolladores de CreateTopic API](#).

Determinación de 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, seleccione la región en la que creó el clúster de MSK.
3. Seleccione 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.

Creación de un tema en la máquina cliente

1. Conéctese a la máquina cliente.
 - a. Abra la consola de Amazon EC2 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 que creó en [Paso 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 mediante la ejecución del siguiente comando.

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

- b. Guarde la [versión de Kafka](#) del clúster de MSK en la variable de entorno, KAFKA_VERSION, tal como se muestra en el siguiente comando. Necesitará esta información durante todo el proceso de configuración.

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

Por ejemplo, si utiliza la versión 3.6.0, use 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
```

Note

La siguiente lista presenta algunas alternativas para la descarga de Kafka que puede utilizar si experimenta algún problema.

- Si experimenta problemas de conectividad o desea utilizar un sitio espejo, pruebe con el selector de espejos de Apache, tal 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 una 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 del directorio recién creado en la variable de entorno KAFKA_ROOT.

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

4. Configure la autenticación para el clúster de MSK.

- a. [Busque la versión más reciente](#) de la biblioteca cliente de IAM para Amazon MSK. Esta biblioteca permite que la máquina cliente acceda al clúster de MSK mediante autenticación con IAM.
- b. Mediante los siguientes comandos, navegue hasta el directorio \$KAFKA_ROOT/libs y descargue el archivo JAR de IAM para Amazon MSK correspondiente que identificó en el paso anterior. Asegúrate de `{LATEST VERSION}` reemplazarlo por el número de versión real que estás descargando.

```
cd $KAFKA_ROOT/libs
```

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

Note

Antes de ejecutar cualquier comando de Kafka que interactúe con el clúster de Amazon MSK, es posible que deba agregar el archivo JAR de IAM de Amazon MSK a la ruta de clases de Java. Establezca la variable de entorno CLASSPATH, tal como se muestra en el siguiente ejemplo.

```
export CLASSPATH=$KAFKA_ROOT/libs/aws-msk-iam-auth-{LATEST VERSION}-all.jar
```

Esto configura CLASSPATH para toda la sesión, lo que hace que el archivo JAR esté disponible para todos los comandos de Kafka posteriores.

- c. Vaya al directorio \$KAFKA_ROOT/config para crear el archivo de configuración del cliente.

```
cd $KAFKA_ROOT/config
```

- d. 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.IAMClientCallbackHandler
```

5. (Opcional) Ajuste el tamaño del montón de Java para las herramientas de Kafka.

Si experimenta problemas relacionados con la memoria o trabaja con una gran cantidad de temas o particiones, puede ajustar el tamaño del montón de Java. Para ello, configure la variable de entorno KAFKA_HEAP_OPTS antes de ejecutar los comandos de Kafka.

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

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

6. Obtenga la información de conexión del clúster.

- Abra la consola de Amazon MSK en <https://console.aws.amazon.com/msk/>.
- 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.
- Seleccione Ver información del cliente.
- Copie la cadena de conexión del punto de conexión privado.

Obtendrá tres puntos de conexión para cada uno de los agentes. Almacene una de estas cadenas de conexión en la variable de entorno BOOTSTRAP_SERVER, como se muestra en el siguiente comando. *<bootstrap-server-string>* Sustitúyalo 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 recibe un error `NoSuchFileException` para el archivo `client.properties`, asegúrese de que este archivo exista en el directorio de trabajo actual, dentro del directorio `bin` de Kafka.

 Note

Si prefiere no establecer la variable de entorno `CLASSPATH` para toda la sesión, puede, como alternativa, anteponer cada comando de Kafka con la variable `CLASSPATH`. Este enfoque aplica la ruta de clases únicamente 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) Verifique que el tema se haya creado correctamente.
 - a. Si el comando se ejecuta correctamente, debería ver el siguiente mensaje: `Created topic MSKTutorialTopic`.
 - b. Enumere todos los temas para confirmar que el tema existe.

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

Si el comando no se ejecuta correctamente o experimenta un error, consulte [Solución de problemas del clúster de Amazon MSK](#) para obtener información de solución de problemas.

9. (Opcional) Elimine las variables de entorno que utilizó en este tutorial.

Si desea conservar las variables de entorno para los siguientes pasos de este tutorial, omita este paso. De lo contrario, puede desestablecer estas 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. Debido a que se trata de una nueva sesión, establezca nuevamente las variables de entorno KAFKA_ROOT y BOOTSTRAP_SERVER. Para obtener información sobre cómo establecer estas variables de entorno, consulte [Creación de un tema en la máquina cliente](#).
4. Ejecute el siguiente comando con la segunda cadena de conexión en 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 comenzar a ver los mensajes que introdujo anteriormente cuando utilizó el comando del productor de consola.

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. Abra 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 Consola de administración de AWS

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. Abra la consola de Amazon EC2 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 completamente administrado que facilita la creación y ejecución de aplicaciones que usan Apache Kafka para procesar datos de streaming. Esta guía proporciona información para ayudar a los desarrolladores a comprender cómo funciona Amazon MSK y cómo usarlo de forma eficaz en sus aplicaciones.

A un nivel general, Amazon MSK proporciona un clúster de Apache Kafka completamente administrado que es aprovisionado y operado por AWS. Esto significa que no se tiene que preocupar por aprovisionar instancias de EC2, configurar ajustes de red, administrar agentes de Kafka ni realizar tareas de mantenimiento continuo. En su lugar, se puede centrar en desarrollar la aplicación y dejar que Amazon MSK se encargue de la infraestructura. Amazon MSK aprovisiona automáticamente los recursos necesarios de computación, almacenamiento y red, y ofrece características como escalado automático, alta disponibilidad y conmutación por error para garantizar que el clúster de Kafka sea confiable y altamente disponible. Esta guía abarca los componentes clave de Amazon MSK y cómo puede usarlo para crear aplicaciones de datos en streaming.

Administración del clúster de MSK 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 [Consola de administración de AWS](#),
- La [referencia de la API de Amazon MSK](#)
- La [referencia de los comandos de la CLI de Amazon MSK](#)

Temas

- [Creación de un clúster de MSK aprovisionado](#)
- [Enumeración de clústeres de Amazon MSK](#)
- [Conexión a un clúster de Amazon MSK aprovisionado](#)
- [Obtención de agentes de arranque para un clúster de Amazon MSK](#)
- [Supervisión de un clúster de Amazon MSK aprovisionado](#)
- [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](#)
- [Aprovisionamiento del rendimiento de almacenamiento para agentes Standard en un clúster de Amazon MSK](#)
- [Actualización del tamaño del agente del clúster de Amazon MSK](#)
- [Utilice LinkedIn el control de cruce 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 de cargas de trabajo de Kafka a un clúster de Amazon MSK](#)
- [Eliminación de un clúster de Amazon MSK aprovisionado](#)

Creación de un clúster de MSK aprovisionado

Important

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

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

Para agentes Standard en la región Oeste de EE: UU. (Norte de California), necesita 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 agentes Express, necesita tres subredes en tres zonas de disponibilidad diferentes. Cuando crea un clúster de MSK aprovisionado, Amazon MSK distribuye los nodos de agentes de manera uniforme entre las subredes que especifique.

Temas

- [Cree un clúster aprovisionado de MSK mediante el Consola de administración de AWS](#)
- [Cree un clúster de Amazon MSK aprovisionado mediante el AWS CLI](#)
- [Cree un clúster aprovisionado de MSK con una configuración de Amazon MSK personalizada mediante el AWS CLI](#)
- [Creación de un clúster de MSK aprovisionado mediante la API de Amazon MSK](#)

Cree un clúster aprovisionado de MSK mediante el Consola de administración de AWS

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

- [Un clúster sin servidor](#)
- [Un clúster de MSK aprovisionado](#) mediante la opción Creación rápida

Procedimientos en este tema

- [Paso 1: Configuración inicial del clúster](#)
- [Paso 2: Configuración del almacenamiento y ajustes del clúster](#)
- [Paso 3: Configuración de los ajustes de red](#)
- [Paso 4: Configuración de los ajustes de seguridad](#)
- [Paso 5: Configuración de las opciones de supervisión](#)
- [Paso 6: Revisión de la configuración del clúster](#)

Paso 1: Configuración inicial 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. En Nombre del clúster, especifique un nombre que sea único y no contenga más de 64 caracteres.
5. En Tipo de clúster, seleccione Aprovisionado.
6. En Versión de Apache Kafka, elija una versión para ejecutar en los agentes. Para ver una comparación de las características de Amazon MSK compatibles con cada versión de Apache Kafka, seleccione Ver compatibilidad de versiones.
7. En la sección Agentes, realice lo siguiente:
 - a. En Tipo de agente, elija una de las siguientes opciones:
 - Express Brokers: High-performance corredores escalables con almacenamiento virtual totalmente gestionado. Elija este tipo de agente para aplicaciones exigentes con requisitos de alto rendimiento.
 - Agentes Standard: agente 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 agente, consulte [Tipos de agentes de Amazon MSK](#).

- b. En Tamaño del agente, elija un tamaño para el clúster en función de las necesidades de computación, memoria y almacenamiento del clúster.
- c. En Número de zonas, elija la cantidad de [Zonas de disponibilidad de AWS](#) entre las cuales se distribuyen los agentes.

Los agentes Express requieren tres zonas de disponibilidad para ofrecer mayor disponibilidad.

- d. En Agentes 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 ZooKeeper-based los clústeres y 60 agentes por clúster para [KRaft-based los clústeres](#).

Paso 2: Configuración del almacenamiento y ajustes del clúster

Este procedimiento describe cómo puede configurar las necesidades de almacenamiento de datos en todos los agentes y especificar el modo de almacenamiento. Esto ayuda a definir los requisitos de almacenamiento de datos en función de las necesidades de la carga de trabajo.

Además, este procedimiento describe las opciones de configuración del clúster que controlan cómo operan los agentes. Estas configuraciones incluyen configuraciones de agentes, configuraciones predeterminadas de temas y la política de almacenamiento por niveles.

1. Si seleccionó el tipo de agente como Standard, realice lo siguiente en la sección Almacenamiento:
 - a. En Almacenamiento, elija la cantidad inicial de almacenamiento que desea que el clúster tenga. Una vez creado el clúster, no se puede reducir la capacidad de almacenamiento.
 - b. (Opcional) Según el tamaño del agente (tamaño de instancia) que haya seleccionado, también puede especificar Rendimiento de almacenamiento aprovisionado por agente. Esta opción le permite asignar un rendimiento de entrada y salida (I/O) dedicado a los volúmenes de Amazon EBS de cada agente.

Para activar esta opción, elija el tamaño del bróker (tamaño de instancia) `kafka.m5.4xlarge` o superior para x86 y `kafka.m7g.2xlarge` o superior para las instancias. Graviton-based A continuación, seleccione la casilla Habilitar rendimiento de almacenamiento aprovisionado. Al seleccionar esta casilla, puede establecer manualmente un mínimo de 250 MiB por segundo de rendimiento. Esto resulta útil para cargas de trabajo o aplicaciones que requieren un rendimiento de almacenamiento predecible y de alta velocidad. I/O-intensive Para obtener más información, consulte [???](#).

- c. En Modo de almacenamiento del clúster, especifique cómo se almacenan y administran los datos dentro del clúster. Esta opción determina el tipo y la configuración del almacenamiento que se utiliza para los agentes. Elija una de las siguientes opciones:
 - Almacenamiento de EBS únicamente: almacena todos los datos de los temas de forma local en volúmenes de Amazon Elastic Block Store (Amazon EBS) conectados a cada agente. Elija este modo para necesidades de rendimiento constante y acceso rápido a mensajes recientes.
 - Almacenamiento por niveles y almacenamiento de EBS: combina datos locales de Amazon EBS con almacenamiento remoto rentable para grandes conjuntos de datos en Amazon S3. Este modo reduce los costos de almacenamiento de Amazon EBS, admite una mayor retención de datos y escala el almacenamiento automáticamente sin intervención manual. Elija este modo cuando desee retener datos durante períodos más prolongados a menor costo o cuando prevea que las necesidades de almacenamiento crecerán de forma significativa.

 Note

No necesita administrar el almacenamiento para los agentes Express.

2. En 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 general. Elija esta opción para una configuración e implementación rápidas del clúster. Para obtener información acerca de las configuraciones de Amazon MSK, consulte [Configuración de Amazon MSK aprovisionado](#).
 - Configuración personalizada: permite especificar configuraciones de agentes y de temas propias. Puede elegir una configuración personalizada existente de la lista o crear una nueva configuración personalizada. Elija esta opción para un control refinado sobre los agentes, como ajustes específicos de rendimiento, configuraciones de seguridad y más.
3. Elija Paso siguiente para continuar.

Paso 3: Configuración de los ajustes de red

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

1. En Redes, realice lo siguiente:
 - a. Elija la VPC que desea utilizar para el clúster.
 - b. Según el número de zonas de disponibilidad que haya seleccionado previamente, especifique las zonas de disponibilidad y las subredes donde se implementarán los agentes.

Para agentes Standard en la región Oeste de EE: UU. (Norte de California), necesita 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 agentes Express, necesita tres subredes en tres zonas de disponibilidad diferentes.

Cuando crea un clúster de MSK aprovisionado, MSK distribuye los nodos de agentes de manera uniforme entre las subredes que especifique.

- c. En Grupos de seguridad en Amazon EC2, elija o cree uno o más grupos de seguridad a los que desee conceder acceso al clúster. Estos grupos de seguridad de Amazon EC2 controlan el tráfico entrante y saliente hacia los 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 [Conexión a un clúster de MSK](#).

2. Elija Paso siguiente para continuar.

Paso 4: Configuración de los ajustes de seguridad

1. En la sección Ajustes de seguridad, realice lo siguiente:

- Elija uno o más de los siguientes métodos de autenticación y autorización para controlar el acceso de los clientes a sus clústeres de Kafka:
 - Acceso no autenticado: permite que los clientes accedan al clúster sin proporcionar credenciales de autenticación. Este método representa un riesgo de seguridad y podría no cumplir 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 IAM. AWS users/roles Este método proporciona un control detallado del acceso al clúster mediante políticas de IAM. Recomendamos este método para las aplicaciones que ya se están ejecutando. AWS
 - SASL/SCRAM autenticación: requiere que los clientes proporcionen las credenciales de nombre de usuario y contraseña almacenadas AWS Secrets Manager para la autenticación. Amazon MSK obtiene estas credenciales desde 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, primero cree un recurso de 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, consulte [Configurar 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 corredores 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 las autoridades de certificación privadas (PCA) solo autenticuen con un único clúster de MSK y así ejercer un control de acceso estricto.

2. En Cifrado, elija el tipo de clave de KMS que desea 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: Configuración de las opciones de supervisión

Este procedimiento describe cómo configurar las métricas de los agentes, así como recopilar y entregar los registros de los agentes. Con estos ajustes, puede observar y analizar el estado del clúster, el rendimiento 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 por clúster sin costo adicional. Este nivel es adecuado para la mayoría de los casos de uso con necesidades generales de supervisión.
 - b. Monitoreo mejorado del agente: proporciona métricas detalladas por agente con un costo adicional. Este nivel incluye una supervisión básica y métricas de los corredores más detalladas, como las métricas de almacenamiento en niveles, los bytes in/out de otros

corredores o el tiempo total de las operaciones. read/write Las métricas de este nivel tienen costo, mientras que las métricas del nivel básico aún son gratuitas.

- c. Monitoreo mejorado del tema: proporciona métricas para temas individuales con un costo adicional. Elija este nivel para obtener una vista más detallada del rendimiento de los temas en los agentes. Este nivel incluye una supervisión mejorada en el nivel de agente y métricas en el nivel de tema, como métricas de almacenamiento por niveles para un tema específico y la cantidad de mensajes recibidos por segundo.
- d. Monitoreo mejorado en el nivel de partición: proporciona la vista más detallada de métricas por partición con un costo adicional. Elija este nivel para obtener la supervisión más detallada mediante la captura de métricas para cada partición en cada tema en los agentes. Este nivel incluye una supervisión mejorada en el nivel de tema y métricas específicas de las particiones detalladas, como las métricas de compensación del retraso.

Para obtener más información sobre las métricas disponibles para los tipos de agente Standard y Express en cada uno de estos niveles de supervisión, consulte [CloudWatch métricas para corredores estándar](#) y [CloudWatch métricas para corredores de Express](#).

2. (Opcional) Si desea exportar métricas en formato Prometheus mediante JMX Exporter, Node Exporter o ambos, seleccione Habilitar supervisión abierta con Prometheus. Para obtener más información acerca de esta opción, consulta [Supervisión con Prometheus](#).
3. (Opcional) Para configurar el clúster de MSK de forma que entregue los registros de los corredores a varios agentes a Servicios de AWS fin de solucionar problemas y realizar auditorías, 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 de la Consola de administración de AWS.
 - Entrega a Amazon S3: almacena los registros como archivos en buckets de Amazon S3 para archivado a largo plazo y análisis 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 ayudar a identificar, organizar o buscar el clúster, seleccione Agregar nueva etiqueta para agregar etiquetas como pares clave-valor. Por ejemplo, agregue una etiqueta al clúster con el par clave-valor de **Load testing y 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: Revisión de la configuración del clúster

1. Revise la configuración del clúster.

Seleccione Editar o Anterior para cambiar cualquiera de las configuraciones que especificó previamente o volver a la pantalla anterior de la consola.

2. Elija Create cluster.
3. Verifique el estado de este clúster en la sección Resumen del clúster de la página 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 [Descripción de los estados de los clústeres de MSK aprovisionados](#).

Cree un clúster de Amazon MSK aprovisionado mediante el AWS CLI

1. Copie el siguiente JSON y guárdelo en un archivo. Nombre el archivo `brokernodegroupinfo.json`. Reemplace los ID de subred en el archivo JSON con los valores que corresponden a las 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 administrar grupos de seguridad de Amazon EC2 asociados con una VPC específica, mediante programación y en la consola](#). Por último, guarde el archivo JSON actualizado en el ordenador en el que lo tiene AWS CLI instalado.

```
{
  "InstanceType": "kafka.m5.large",
  "ClientSubnets": [
    "Subnet-1-ID",
```

```

    "Subnet-2-ID"
  ],
  "SecurityGroups": [
    "Security-Group-ID"
  ]
}

```

Important

Para agentes Express, necesita tres subredes en tres zonas de disponibilidad diferentes. Tampoco necesita definir ninguna propiedad relacionada con el almacenamiento. Para agentes Standard en la región Oeste de EE: UU. (Norte de California), necesita 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. Ejecuta el siguiente AWS CLI comando en el directorio donde guardaste el `brokernodegroupinfo.json` archivo y *"Your-Cluster-Name"* sustitúyelo por el nombre que prefieras. 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 [Descripción de los estados de los clústeres de MSK aprovisionados](#).

```
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 “Configuraciones de agentes”](#).

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"  
}
```

Creación de un clúster de MSK provisionado mediante la API de Amazon MSK

La API de Amazon MSK permite crear y administrar mediante programación el clúster de MSK provisionado como parte de procesos automatizados de aprovisionamiento de infraestructura o de scripts de implementación.

Para crear un clúster provisionado 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 “Obtención de los agentes de arranque”](#).

Temas

- [Enumere los clústeres mediante la Consola de administración de AWS](#)
- [Enumere los clústeres mediante el AWS CLI](#)
- [Enumeración de clústeres mediante la API](#)

Enumere los clústeres mediante la Consola de administración de AWS

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 “Obtención de los agentes de arranque”](#).

1. [Inicie sesión en y abra la Consola de administración de AWS consola de Amazon MSK en?](https://console.aws.amazon.com/msk/home region=us-east-1#/home/) <https://console.aws.amazon.com/msk/home 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 “Obtención de los agentes de arranque”](#).

```
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 “Obtención de los agentes de arranque”](#).

Para enumerar los clústeres que utilizan la API, consulte [ListClusters](#).

Conexión a un clúster de Amazon MSK aprovisionado

De forma predeterminada, los clientes solo pueden acceder a un clúster de MSK aprovisionado si se encuentran en la misma VPC que el clúster. Toda la comunicación entre los clientes de Kafka y el clúster de MSK aprovisionado es privada de forma predeterminada, y los datos de streaming nunca atraviesan Internet. Para conectarse al clúster de MSK aprovisionado desde un cliente que se encuentra en la misma VPC que el clúster, asegúrese de que el grupo de seguridad del clúster tenga una regla de entrada que permita el tráfico desde el 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 instancia de Amazon EC2 que esté en la misma VPC que el clúster, consulte [the section called “Introducción”](#).

Note

KRaft el modo de metadatos y los corredores de MSK Express no pueden tener habilitados la supervisión abierta y el acceso público.

Para conectarte a tu clúster aprovisionado de MSK desde un cliente que está fuera de la VPC del clúster, consulta [Acceso desde dentro AWS pero desde fuera de la VPC del clúster](#).

Temas

- [Activación del acceso público a un clúster de MSK aprovisionado](#)
- [Acceso desde la AWS VPC del clúster pero desde fuera](#)

Activación del acceso público a un clúster de MSK aprovisionado

Amazon MSK ofrece la opción de habilitar el acceso público a los agentes de clústeres de MSK aprovisionados que ejecutan Apache Kafka versión 2.6.0 o posterior. 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.

Puedes activar el acceso público a un clúster de MSK sin coste adicional, pero se aplican los costes de transferencia de AWS datos estándar para la transferencia de datos dentro y fuera del clúster. Para obtener más información sobre este modelo de precios, consulte [Precios de las instancias bajo demanda de Amazon EC2](#).

Los clústeres aprovisionados de Amazon MSK con un tipo de red de doble pila admiten ambos IPv4 tipos de IPv6 conectividad para el acceso público. Cuando el acceso público esté habilitado en su clúster, las mismas cadenas de IPv6 arranque funcionarán automáticamente tanto para la conectividad predeterminada como para la de acceso público. Las cadenas de IPv4 arranque existentes seguirán funcionando para IPv4 la conectividad. Tenga en cuenta que si el acceso público no está habilitado en su clúster, las cadenas de IPv6 arranque no tendrán capacidad de acceso público. Para obtener más información, consulte [Configurar el tipo de red de doble pila para un clúster de Amazon MSK](#).

Note

Si utiliza los métodos de control de acceso SASL/SCRAM o mTLS, primero debe configurar Apache Kafka para su clúster. ACLs Luego, actualice la configuración del clúster para establecer la propiedad `allow.everyone.if.no.acl.found` en falso. Para obtener información acerca de cómo actualizar la configuración de un clúster, consulte [the section called “Operaciones de configuración de agentes”](#).

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

- 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 información sobre cómo crear y asociar una puerta de enlace de Internet, consulte [Habilitación del acceso a VPC mediante puertas de enlace de Internet](#) en la Guía del usuario de Amazon VPC.
- El control de acceso no autenticado debe estar desactivado 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 “Actualización de 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 “Actualización de la seguridad del clúster”](#).
- Si utiliza control de acceso con IAM y desea aplicar o actualizar políticas de autorización, consulte [the section called “Control de acceso de IAM”](#). Para obtener información sobre Apache Kafka, consulte ACLs [the section called “ACL de Apache Kafka”](#).

Una vez que se asegure de que un clúster de MSK cumple las condiciones enumeradas anteriormente, puede usar la Consola de administración de AWS, 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 “Obtención de los agentes de arranque”](#).

 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 Consola de administración de AWS 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 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.

Activar el acceso público mediante el AWS CLI

1. Ejecute el siguiente AWS CLI comando, sustituyendo *ClusterArn* y *Current-Cluster-Version* por el ARN y la versión actual del clúster. Para encontrar la versión actual del clúster, utilice la [DescribeCluster](#) operación o el comando [AWS CLI describe-cluster](#). Un ejemplo de ID de versión es KTVDPKIKX0DER.

```
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, usa un AWS CLI comando 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 y `ClusterOperationArn` reemplázelo por el ARN que obtuvo en el resultado 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 ni a los nodos KRaft del controlador.

Acceso desde la AWS VPC del clúster pero desde fuera

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

Interconexión de Amazon VPC

Para conectarte a tu clúster de MSK desde una VPC diferente de la VPC del clúster, puedes crear una conexión de emparejamiento entre ambas. VPCs Para obtener información sobre las interconexiones de VPC, consulte la [Guía de interconexión de Amazon VPC](#).

Direct Connect

Direct Connect conecta la red local a AWS través de un cable de fibra óptica Ethernet estándar de 1 o 10 gigabits. Un extremo del cable está conectado al router y el otro a un router. Direct Connect

Con esta conexión, puede crear interfaces virtuales directamente a la AWS nube y a Amazon VPC, sin tener en cuenta a los proveedores de servicios de Internet en su ruta de red. Para obtener más información, consulte [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

El siguiente documento describe las opciones de conectividad para varias VPCs que residen en diferentes regiones: Conectividad [multiVPC de varias regiones](#).

Conectividad privada con varias VPC en una sola región

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

Consulte [Single Region multi-VPC connectivity for cross-account clients](#).

Fin de la compatibilidad con la red de EC2-Classic

Amazon MSK ya no es compatible con las instancias de Amazon EC2 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 multiVPC (con tecnología [AWS PrivateLink](#)) para los clústeres de Amazon Managed Streaming for Apache Kafka (Amazon MSK) es una función que le permite conectar más rápidamente los clientes de Kafka alojados en diferentes VPCs 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 PrivateLink mientras mantienen todo el tráfico dentro de la AWS red. La conectividad privada de múltiples VPC para los clústeres de Amazon MSK está disponible en todas las regiones en las que está disponible Amazon MSK.

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 de múltiples VPC para Amazon MSK es una opción de conectividad que le permite conectar clientes de Apache Kafka que están alojados en diferentes redes privadas virtuales (VPCs) y AWS cuentas 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 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 la superposición IPs entre las conexiones VPCs, lo que elimina la necesidad de mantener tablas de enrutamiento y IPs emparejamiento complejas y que no se superpongan 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 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 utilizas los métodos de control de acceso SASL/SCRAM o mTLS, debes configurar Apache Kafka para tu 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 agentes”](#). 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, consulte [the section called “ACL de Apache Kafka”](#).
- La conectividad privada con varias VPC no admite el tipo de instancia `t3.small`.
- La conectividad privada de múltiples VPC no se admite en todas AWS las regiones, solo en AWS las cuentas de la misma región.
- Para configurar la conectividad privada entre múltiples VPC, debe contar con el mismo número de subredes de cliente que de subredes del clúster. También debe asegurarse de que la [zona IDs de disponibilidad](#) sea la misma para la 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 de varias VPC para conectar de forma privada un cliente de Apache Kafka a un clúster de MSK desde dentro, 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.

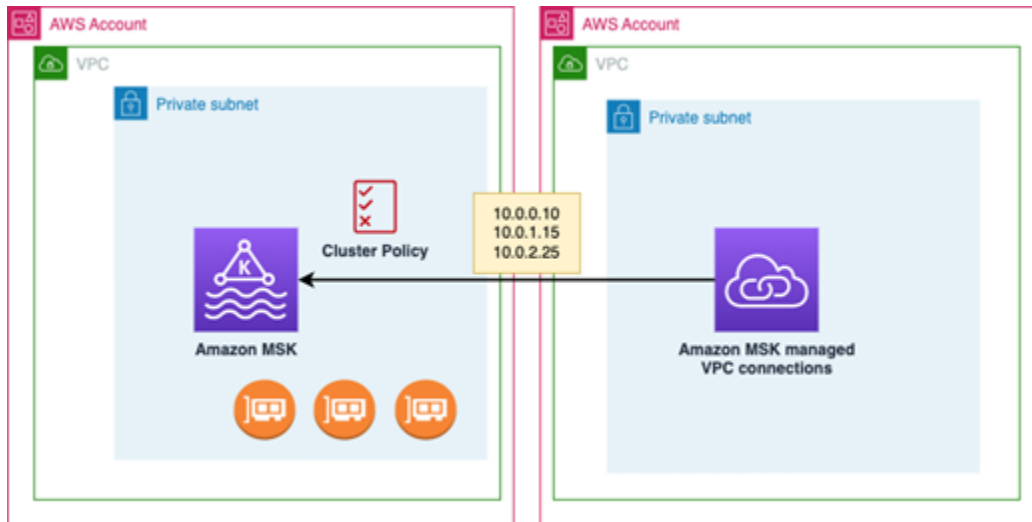
La configuración de red para este caso de uso es la siguiente:

- 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 multiVPC de Amazon MSK conectada a un cliente de una cuenta 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 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 el clúster de MSK para que utilice un esquema de SASL/SCRAM autenticación, la ACLs propiedad "" allow.everyone.if.no.acl.found=false de Apache Kafka es obligatoria. [Consulte Apache Kafka. ACLs](#)

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 encuentra 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 de varias VPC en un clúster

Como alternativa a la consola de MSK, puede usar la [UpdateConnectivity API](#) para activar la conectividad privada de 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 de acceso al clúster, a los temas y a los grupos.

JSON

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

```

{
  "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 **111122223333**.
5. Seleccione Guardar cambios.

Para obtener información sobre la política de clústeres APIs, consulte Políticas [basadas en recursos de Amazon MSK](#).

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.

JSON

```
{
  "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 [DescribeCluster](#) API 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 [GetBootstrapBrokers](#) API 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 arranque para conectarse de forma privada al clúster de Amazon MSK.

Actualización de los esquemas de autorización de un clúster

La conectividad privada de múltiples 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 [Consola de administración de AWS](#) 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 [Consola de administración de AWS](#).
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 Para convertir on/off un esquema de autenticación con la API

Como alternativa a la consola de MSK, puede usar la [UpdateConnectivity API](#) para activar la conectividad privada de varias VPC y configurar esquemas de autenticación en un clúster ACTIVO. En el siguiente ejemplo, se muestran los esquemas SASL/SCRAM de autenticación de 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",
  "connectivityInfo": {
    "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 [Consola de administración de AWS](#).
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 [Consola de administración de AWS](#).
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 el clúster de MSK para que utilice un esquema de SASL/SCRAM autenticación, la propiedad de Apache Kafka ACLs es obligatoria. `allow.everyone.if.no.acl.found=false` Después de configurar las [ACL de Apache Kafka](#) 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 de agentes](#).

Permisos de políticas de clúster entre cuentas

Si un cliente de Kafka está en una AWS cuenta diferente a la del clúster de MSK, adjunte al clúster de MSK una política basada en el clúster que autorice al usuario raíz del cliente a conectarse entre cuentas. Puede editar la política de clúster de varias VPC mediante el editor de políticas de IAM de la consola de MSK (Configuración de seguridad del clúster > Editar política de 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 de acceso al clúster, a los temas y a los grupos.

JSON

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

```
}
```

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.

JSON

```
{
  "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 intermediarios mediante el cifrado TLS, utilice el puerto 9094 para acceder desde dentro AWS y el puerto 9194 para el acceso público.
- Para comunicarse con los intermediarios mediante SASL/SCRAM, utilice el puerto 9096 para acceder desde dentro AWS y el puerto 9196 para el acceso público.

- Para comunicarse con los corredores de un clúster configurado para su uso [the section called “Control de acceso de IAM”](#), utilice el puerto 9098 para acceder desde dentro AWS y el puerto 9198 para el acceso público.
- Para comunicarse con los corredores mediante el tipo de IPv6 red en texto sin formato, utilice el puerto 20092
- Para comunicarse con los corredores de un clúster que está configurado para utilizar el control de acceso de IAM IPv6, utilice el puerto 20098.
- Para comunicarse con los corredores y utilizarlos IPv6, SASL/SCRAM utilice el puerto 20096.
- Para comunicarse con los corredores que utilizan el cifrado TLS IPv6, utilice el puerto 20094.

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 Consola de administración de AWS AWS CLI, o Amazon MSK.

Temas

- [Obtenga los corredores de bootstrap utilizando la Consola de administración de AWS](#)
- [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 Consola de administración de AWS

Este proceso describe cómo obtener corredores de arranque para un clúster utilizando el. Consola de administración de AWS 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. [Inicie sesión en y abra la Consola de administración de AWS consola de Amazon MSK en? https://console.aws.amazon.com/msk/home region=us-east-1#/home/](https://console.aws.amazon.com/msk/home 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](#).

Supervisión de un clúster de Amazon MSK aprovisionado

Amazon MSK ofrece varias formas de ayudar a supervisar el estado del clúster de Amazon MSK aprovisionado.

- 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 aprender cómo supervisar el clúster de MSK aprovisionado con Prometheus, consulte [the section called “Supervisión con Prometheus”](#).
- (Solo agentes Standard) Amazon MSK ayuda a supervisar la capacidad de almacenamiento en disco mediante el envío automático de alertas de capacidad de almacenamiento cuando un clúster aprovisionado está a punto de alcanzar su límite de capacidad. 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 Panel de estado 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](#)
- [Supervisión de un clúster de MSK aprovisionado 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 Consola de administración de AWS 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 los comandos [list-metrics](#) y [get-metric-statistics](#).

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 de MSK aprovisionado en uno de los siguientes: DEFAULT, PER_BROKER, PER_TOPIC_PER_BROKER o

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](#).

Nivel de monitoreo **PREDETERMINADO**

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

Name	Cuando está visible	Dimensiones	Description (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

Name	Cuando está visible	Dimensiones	Description (Descripción)
			más información, consulte Rendimiento de I/O créditos y 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, ID del agente, Tema	El número de bytes por segundo enviados a los clientes. Esta métrica 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.

Name	Cuando está visible	Dimensiones	Description (Descripción)
CPUCredit Balance	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.

Name	Cuando está visible	Dimensiones	Description (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 .
KafkaApplicationDiskUsed	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.

Name	Cuando está visible	Dimensiones	Description (Descripción)
KafkaData LogsDiskUsed (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..

Name	Cuando está visible	Dimensiones	Description (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.
NetworkRx Dropped	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de paquetes abandonados descartados.
NetworkRx Errors	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.

Name	Cuando está visible	Dimensiones	Description (Descripción)
NetworkRx Packets	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de paquetes recibidos por el agente.
NetworkTx Dropped	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de paquetes de transmisión descartados.
NetworkTx Errors	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.
NetworkTx Packets	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.

Name	Cuando está visible	Dimensiones	Description (Descripción)
ProduceTo taTimeMsMean	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El tiempo medio de producción en milisegundos.
RequestBy tesMean	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 dedica a la red de intermediarios y a I/O los subprocesos a procesar las 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.
RollingEs timatedTi meLagMax*	Después de que el grupo de consumidores consuma de un tema.	Nombre del clúster, grupo de consumidores, tema	Estimación del tiempo máximo acumulativo (en segundos) para reducir el desfase de compensación de la partición en todas las particiones de un tema.

Name	Cuando está visible	Dimensiones	Description (Descripción)
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.
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	High-level métricas que indican la cantidad de paquetes moldeados (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.

Name	Cuando está visible	Dimensiones	Description (Descripción)
UnderReplicatedPartitions	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	El número de particiones infrareplicadas para el agente.
UserPartitionExists	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	Una métrica booleana que indica la presencia de una partición propiedad del usuario en un agente. Un valor de 1 indica la presencia de particiones en el agente.
ZooKeeperRequestLatencyMsMean	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster, ID del agente	Para ZooKeeper-based clúster. 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 el ZooKeeper-based clúster. El estado de conexión de la ZooKeeper sesión del bróker puede ser uno de los siguientes: NOT_CONNECTED: '0.0', ASSOCIATING: '0.1', CONNECTEDREADONLY: '0.8', CONNECTED: '1.0', CLOSED: '5.0', AUTH_FAILED: '10.0'.

* Los indicadores de retardo de consumo requieren nombres de grupos de consumidores y tienen requisitos de emisiones específicos. ASCII-only Para obtener más información, consulte [the section called “Supervisión del desfase del consumidor”](#).

PER_BROKER: Monitorización de niveles

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.

Name	Cuando está visible	Description (Descripción)
BwInAllowanceExceeded	Después de que el clúster llegue al estado ACTIVE (Activo).	El número de paquetes formados porque el ancho de banda agregado entrante superó el máximo del agente.
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.
CpuCreditUsage	Después de que el clúster llegue	La cantidad de créditos de CPU empleados por el agente. Si se

Name	Cuando está visible	Description (Descripción)
	al estado ACTIVE (Activo).	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 producer/consumer	Tiempo medio en milisegundos que la solicitud del consumidor se procesa en el líder.
FetchConsumerRequestQueueTimeMsMean	Después de que haya un producer/consumer.	Tiempo medio en milisegundos que la solicitud del consumidor espera en la cola de solicitudes.
FetchConsumerResponseQueueTimeMsMean	Después de que haya un producer/consumer.	Tiempo medio en milisegundos que la solicitud del consumidor espera en la cola de respuesta.
FetchConsumerResponseSendTimeMsMean	Después de que haya un producer/consumer.	Tiempo medio en milisegundos para que el consumidor envíe una respuesta.
FetchConsumerTotalTimeMsMean	Después de que haya un producer/consumer.	El tiempo total medio en milisegundos que los consumidores gastan en obtener datos del agente.
FetchFollowerLocalTimeMsMean	Después de que haya un producer/consumer.	Tiempo medio en milisegundos que la solicitud del seguidor se procesa en el líder.
FetchFollowerRequestQueueTimeMsMean	Después de que haya un producer/consumer.	Tiempo medio en milisegundos que la solicitud del seguidor espera en la cola de solicitudes.

Name	Cuando está visible	Description (Descripción)
FetchFollowerResponseQueueTimeMsMean	Después de que haya un productor/consumer.	Tiempo medio en milisegundos que la solicitud del seguidor espera en la cola de respuesta.
FetchFollowerResponseSendTimeMsMean	Después de que haya un productor/consumer.	Tiempo medio en milisegundos para que el seguidor envíe una respuesta.
FetchFollowerTotalTimeMsMean	Después de que haya un productor/consumer.	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.

Name	Cuando está visible	Description (Descripción)
IAMTooManyConnections	Después de que el clúster llegue al estado ACTIVE (Activo).	La cantidad de conexiones intentada s 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.
LinklocalAllowanceExceeded	Después de que el clúster llegue al estado ACTIVE (Activo).	El número de paquetes eliminados porque el PPS del tráfico a los servicios proxy locales superó el máximo para la interfaz de red. Esto afecta al tráfico hacia el servicio de DNS, el servicio de metadatos de instancia y el servicio de sincronización temporal de Amazon.
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.

Name	Cuando está visible	Description (Descripción)
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.
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.

Name	Cuando está visible	Description (Descripción)
RemoteFetchBytesPerSec (RemoteBytesInPerSec in v2.8.2.tiered)	Después de que haya un producer/consumer.	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. Esta es una KIP-405 métrica.
RemoteCopyBytesPerSec (RemoteBytesOutPerSec in v2.8.2.tiered)	Después de que haya un producer/consumer.	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. Esta es una KIP-405 métrica.
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. Es una KIP-405 métrica.

Name	Cuando está visible	Description (Descripción)
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. Es una KIP-405 métrica.
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. Es una KIP-405 métrica.
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. Es una KIP-405 métrica.

Name	Cuando está visible	Description (Descripción)
RemoteFetchRequestPerSec (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. Es una KIP-405 métrica.
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. Es una KIP-405 métrica.
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 clústeres con almacenamiento por niveles a partir 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.

Name	Cuando está visible	Description (Descripción)
<code>RequestExemptFromThrottleTime</code>	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.
<code>RequestHandlerAvgIdlePercent</code>	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.
<code>RequestThrottleQueueSize</code>	Después de aplicar la limitación de solicitudes.	El número de mensajes en la cola del acelerador.
<code>RequestThrottleTime</code>	Después de aplicar la limitación de solicitudes.	El tiempo medio de aceleración de la solicitud en milisegundos.
<code>TcpConnections</code>	Después de que el clúster llegue al estado ACTIVE (Activo).	Muestra el número de segmentos TCP entrantes y salientes con el indicador SYN establecido.
<code>RemoteCopyLagBytes (TotalTierBytesLag in v2.8.2.tiered)</code>	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 KIP-405 métrica.

Name	Cuando está visible	Description (Descripción)
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.
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.

Name	Cuando está visible	Description (Descripción)
VolumeWriteOps	Después de que el clúster llegue al estado ACTIVE (Activo).	El número total de operaciones de escritura en un periodo especificado.

PER_TOPIC_PER_BROKER Supervisión de niveles

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.

Name	Cuando está visible	Description (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 (RemoteBy	Después de crear un tema, el tema es.	El número de bytes transferidos desde el almacenamiento por niveles en respuesta a las búsquedas del consumidor por el tema y

Name	Cuando está visible	Description (Descripción)
<code>tesInPerSec in v2.8.2.tiered)</code>	producing/ consuming	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 KIP-405 métrica.
<code>RemoteCopyBytesPerSec (RemoteBytesOutPerSec in v2.8.2.tiered)</code>	Después de crear un tema y el tema es producing/ consuming.	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 KIP-405 métrica.
<code>RemoteFetchErrorsPerSec (RemoteReadErrorPerSec in v2.8.2.tiered)</code>	Después de crear un tema y el tema es producing/ consuming.	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. Categoría: tasas de tráfico y error. Se trata de una KIP-405 métrica.
<code>RemoteFetchRequestPerSec (RemoteReadRequestsPerSec in v2.8.2.tiered)</code>	Después de crear un tema y el tema es producing/ consuming.	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 KIP-405 métrica.

Name	Cuando está visible	Description (Descripción)
RemoteCopyErrorsPerSec (RemoteWriteErrorPerSec in v2.8.2.tiered)	Después de crear un tema y el tema es producing/consuming.	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 KIP-405 métrica.
RemoteLogSizeBytes	Después de crear un tema.	El número de bytes almacenados en el nivel remoto. Esta métrica está disponible para clústeres con almacenamiento por niveles a partir de Apache Kafka versión 3.7.x en Amazon MSK.

PER_TOPIC_PER_PARTITION Supervisión del nivel

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.

Name	Cuando está visible	Description (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.

Name	Cuando está visible	Description (Descripción)
OffsetLag*	Después de que el grupo de consumidores consuma de un tema.	Partition-level retraso del consumidor en el número de compensaciones.
RollingEstimatedTimeLag*	Después de que el grupo de consumidores consuma de un tema.	Estimación del tiempo de rotación (en segundos) para reducir el retardo de compensación de la partición.

* Las métricas de retardo de ASCII-only consumo requieren los nombres de los grupos de consumidores y tienen requisitos de emisiones específicos. Para obtener más información, consulte [the section called “Supervisión del desfase del consumidor”](#).

Descripción de los estados de los clústeres de MSK aprovisionados

La siguiente tabla muestra los posibles estados de un clúster de MSK aprovisionado y describe su significado. Salvo que se indique lo contrario, los estados de los clústeres de MSK aprovisionados se aplican tanto a los tipos de agente Standard como Express. Esta tabla también describe qué acciones puede y no puede realizar cuando un clúster de MSK aprovisionado se encuentra en uno de estos estados. Para conocer el estado de un clúster, puede visitar la Consola de administración de AWS. También puede usar el comando [describe-cluster-v2](#) o la operación V2 para describir el clúster [DescribeClusteraprovisionado](#). La descripción de un clúster incluye su estado.

Estado del clúster de MSK aprovisionado	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 en proceso de configurar el clúster aprovisionado. Debe esperar a que

Estado del clúster de MSK aprovisionado	Significado y posibles acciones
	el clúster 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	El clúster aprovisionado está en proceso de ser eliminado. No puede usarlo para producir o consumir datos. Tampoco puede realizar AWS CLI operaciones ni la API de Amazon MSK en ella.
ERROR	El proceso de creación o eliminación del clúster aprovisionado falló. 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. Aún puede usar 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 de MSK aprovisionado	Significado y posibles acciones
MAINTENANCE	(Solo agentes Standard) Amazon MSK realiza operaciones de mantenimiento rutinarias en el clúster. Estas operaciones de mantenimiento 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 agentes Express. Consulte Aplicación de parches en clústeres de MSK aprovisionados.
REBOOTING_BROKER	Amazon MSK está reiniciando un agente. Aún puede usar 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 actualizando el clúster aprovisionado. Aún puede usar 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 por MSK se recopilan automáticamente y se actualizan CloudWatch en intervalos de 1 minuto. Puede establecer el nivel de supervisión de un clúster de MSK aprovisionado en uno de los siguientes: DEFAULT, PER_BROKER, PER_TOPIC_PER_BROKER o PER_TOPIC_PER_PARTITION. Las tablas de las secciones siguientes muestran las métricas que están 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](#).

Monitoreo de nivel **PREDETERMINADO** para corredores de Express

Las métricas descritas en la tabla siguiente están disponibles sin costo en el nivel de supervisión DEFAULT.

Name	Cuando está visible	Dimensiones	Description (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 disponible

Name	Cuando está visible	Dimensiones	Description (Descripción)
			e 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.
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.

Name	Cuando está visible	Dimensiones	Description (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. Debido a que GlobalPartitionCount no incluye réplicas, la suma de los valores de PartitionCount 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.	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.

Name	Cuando está visible	Dimensiones	Description (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 o descartados.

Name	Cuando está visible	Dimensiones	Description (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.

Name	Cuando está visible	Dimensiones	Description (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 corredores y en I/O los subprocesos para procesar las solicitudes.
RollingEstimatedTimeLagMax*	Después de que el grupo de consumidores consuma de un tema.	Grupo de consumidores, tema	Estimación del tiempo máximo acumulativo (en segundos) para reducir el desfase de compensación de la partición en todas las particiones de un tema.
StorageUsed	Después de que el clúster llegue al estado ACTIVE (Activo).	Nombre del clúster	El almacenamiento total utilizado en todas las particiones del clúster, con exclusión de las réplicas.
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.

Name	Cuando está visible	Dimensiones	Description (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 agente. Un valor de 1 indica la presencia de particiones en el agente.

* Los indicadores de retardo de ASCII-only consumo requieren el nombre de los grupos de consumidores y tienen requisitos de emisiones específicos. Para obtener más información, consulte [the section called “Supervisión del desfase del consumidor”](#).

PER_BROKER: Monitorización a nivel 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. Paga por las métricas de la tabla siguiente, mientras que las métricas del nivel DEFAULT se mantienen gratuitas. Las métricas de esta tabla tienen las dimensiones siguientes: nombre del clúster, identificador del agente.

Name	Cuando está visible	Description (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 conexiones nuevas establecidas por segundo por oyente. Este número se agrega por oyente y se filtra para los oyentes del cliente.

Name	Cuando está visible	Description (Descripción)
FetchConsumerLocalTimeMsMean	Después hay un. producer/consumer	Tiempo medio en milisegundos que la solicitud del consumidor se procesa en el líder.
FetchConsumerRequestQueueTimeMsMean	Después de que haya un producer/consumer.	Tiempo medio en milisegundos que la solicitud del consumidor espera en la cola de solicitudes.
FetchConsumerResponseQueueTimeMsMean	Después de que haya un producer/consumer.	Tiempo medio en milisegundos que la solicitud del consumidor espera en la cola de respuesta.
FetchConsumerResponseSendTimeMsMean	Después de que haya un producer/consumer.	Tiempo medio en milisegundos para que el consumidor envíe una respuesta.
FetchConsumerTotalTimeMsMean	Después de que haya un producer/consumer.	El tiempo total medio en milisegundos que los consumidores gastan en obtener datos del agente.
FetchFollowerLocalTimeMsMean	Después de que haya un producer/consumer.	Tiempo medio en milisegundos que la solicitud del seguidor se procesa en el líder.
FetchFollowerRequestQueueTimeMsMean	Después de que haya un producer/consumer.	Tiempo medio en milisegundos que la solicitud del seguidor espera en la cola de solicitudes.

Name	Cuando está visible	Description (Descripción)
FetchFollowerResponseQueueTimeMsMean	Después de que haya un producer/consumer.	Tiempo medio en milisegundos que la solicitud del seguidor espera en la cola de respuesta.
FetchFollowerResponseSendTimeMsMean	Después de que haya un producer/consumer.	Tiempo medio en milisegundos para que el seguidor envíe una respuesta.
FetchFollowerTotalTimeMsMean	Después de que haya un producer/consumer.	El tiempo total medio en milisegundos que los seguidores gastan en obtener datos del 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.

Name	Cuando está visible	Description (Descripción)
IAMTooManyConnections	Después de que el clúster llegue al estado ACTIVE (Activo).	El número de conexiones intentadas por encima de 100. 0 significa que el número de conexiones se encuentra dentro del límite. Si >0, el límite de limitación controlada se está superando y debe 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.
ProduceThrottleQueueSize	Después de aplicar la limitación del ancho de banda.	El número de mensajes en la cola del acelerador.

Name	Cuando está visible	Description (Descripción)
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 corredores y a I/O los subprocesos para 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 solicitud 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.

Name	Cuando está visible	Description (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.

Supervisión del nivel **PER_TOPIC_PER_PARTITION** para **corredores** de Express

Cuando establece el nivel de supervisión en **PER_TOPIC_PER_PARTITION**, obtiene 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 del nivel **DEFAULT** son gratuitas. Las métricas de esta tabla tienen las siguientes dimensiones: grupo de consumidores, tema y partición.

Name	Cuando está visible	Description (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.	Partition-level retraso con respecto al consumo en el número de compensaciones.
RollingEstimatedTimeLag*	Después de que el grupo de consumidores consuma de un tema.	Estimación del tiempo de rotación (en segundos) para reducir el retardo de compensación de la partición.

* Las métricas de retardo de ASCII-only consumo requieren los nombres de los grupos de consumidores y tienen requisitos de emisiones específicos. Para obtener más información, consulte [the section called “Supervisión del desfase del consumidor”](#).

Monitorización a **nivel PER_TOPIC_PER_BROKER** para corredores de Express

Cuando establece el nivel de supervisión en PER_TOPIC_PER_BROKER, obtiene las métricas descritas en la tabla siguiente, además de todas las métricas de los niveles PER_BROKER y DEFAULT. Solo las métricas del nivel DEFAULT son gratuitas. Las métricas de esta tabla tienen las dimensiones siguientes: nombre del clúster, identificador del agente, tema.

 Important

Las métricas de la tabla siguiente aparecen únicamente 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.

Name	Cuando está visible	Description (Descripción)
MessagesInPerSec	Después de crear un tema.	El número de mensajes recibidos por segundo.

Supervisión de un clúster de MSK provisionado con Prometheus

Puede supervisar el clúster de MSK provisionado con Prometheus, un sistema de supervisión de código abierto para datos de métricas de series de tiempo. 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 Prometheus-formatted las métricas 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 más información sobre el uso de Prometheus, consulte [Cómo mejorar la visibilidad operativa de Amazon MSK mediante Amazon Managed Service para Prometheus y Amazon Managed Grafana](#).

 Note

En el modo de metadatos KRaft, los agentes Express de MSK no pueden tener habilitados simultáneamente la supervisión abierta y el acceso público.

Habilitación de la supervisión abierta en nuevos clústeres de MSK aprovisionados

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

Uso de Consola de administración de AWS

1. [Inicie sesión en y abra la Consola de administración de AWS consola de Amazon MSK en? https://console.aws.amazon.com/msk/home?region=us-east-1#/home/.](https://console.aws.amazon.com/msk/home?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.

Uso de 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 operación y especifique. [CreateCluster](#)OpenMonitoring Habilite jmxExporter, nodeExporter, o ambos. Si especifica OpenMonitoring, no se pueden desactivar los dos exportadores al mismo tiempo.

Habilitación de la supervisión abierta en un clúster de MSK aprovisionado existente

Para habilitar la supervisión abierta, asegúrese de que el clúster de MSK aprovisionado se encuentre en el estado ACTIVE.

Uso de Consola de administración de AWS

1. [Inicie sesión en y abra la Consola de administración de AWS consola de Amazon MSK en? https://console.aws.amazon.com/msk/home region=us-east-1#/home/](https://console.aws.amazon.com/msk/home 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. Elija Edit (Edición de).
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).

Uso de 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 operación y especifique. [UpdateMonitoring](#)OpenMonitoring Habilite jmxExporter, nodeExporter, o ambos. Si especifica OpenMonitoring, no se pueden desactivar los dos exportadores al mismo tiempo.

Configuración de un host de Prometheus en una instancia de Amazon EC2

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

1. Descargue el servidor de Prometheus de <https://prometheus.io/download/#prometheus> a su instancia de 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 [ListNodes](#) operación para obtener una lista de los agentes de su clúster.
5. Cree un archivo llamado `targets.json` con el siguiente JSON. Sustituya *broker_dns_1* *broker_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 los controladores de KRaft, agregue los nombres de DNS de los controladores como destinos 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 Amazon EC2, ejecute el siguiente comando en el directorio donde extrajo los archivos de Prometheus y guardó `prometheus.yml` y `targets.json`.

```
./prometheus
```

7. Busque la dirección IP pública IPv4 de la instancia de Amazon EC2 en la que ejecutó 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, abra un navegador que pueda acceder a su instancia de Amazon EC2 y vaya *Prometheus-Instance-Public-IP:9090* a, *Prometheus-Instance-Public-IP* donde está la dirección IP pública que obtuvo 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 desfase del consumidor también están disponibles en el puerto 11001 con el nombre MBean de JMX `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 for Prometheus es Prometheus-compatible un servicio de supervisión y alertas que puede utilizar para supervisar los 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: `EstimatedMaxTimeLag`, `EstimatedTimeLag`, `MaxOffsetLag`, `OffsetLag` y `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 desfase del consumidor se emiten únicamente si un grupo de consumidores se encuentra 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 desfase del consumidor no se emiten en los siguientes escenarios:
 - Si el grupo de consumidores es inestable.
 - Si el nombre del grupo de consumidores contiene dos puntos (:).

- Si no ha configurado el desplazamiento del consumidor 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 UTF-8 caracteres 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, se eliminan las métricas de retardo CloudWatch 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 del 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 [ListClusters](#) la API [ListClustersV2](#) o [DescribeClusterV2](#) para ver todas CustomerActionStatus las alertas 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 dirigir 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 [UpdateSecurity](#) Amazon 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 en el clúster (de agente a agente).

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	Client-broker opciones de cifrado	Broker-broker cifrado
Unauthenticated	TLS, PLAINTEXT, TLS_PLAINTEXT	Puede estar activado o desactivado.

Autenticación	Client-broker opciones de cifrado	Broker-broker cifrado
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 el Consola de administración de AWS

1. [Inicie sesión en y abra la Consola de administración de AWS consola de Amazon MSK en? https://console.aws.amazon.com/msk/home region=us-east-1#/home/](https://console.aws.amazon.com/msk/home 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 dentro del clúster (de agente a agente).

```
{"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 el Consola de administración de AWS

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

1. [Inicie sesión en y abra la Consola de administración de AWS consola de Amazon MSK en?](https://console.aws.amazon.com/msk/home region=us-east-1#/home/)
2. Elija el clúster de MSK para aumentar su número de agentes.
3. En el menú desplegable Acciones, elija Editar número de agentes.
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 el 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 encontrar la versión actual del clúster, utilice la [DescribeCluster](#) operación o el comando [AWS CLI describe-cluster](#). Un ejemplo de ID de versión es KTVDPKIKX0DER.

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 agentes 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 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.

La eliminación de agentes está permitida en las versiones 2.8.1 y posteriores de Kafka, incluidos los clústeres en modo KRaft.

Temas

- [Prepárese para eliminar los agentes quitando todas las particiones](#)
- [Elimine un agente con el AWS Consola de administración](#)
- [Elimine un corredor con la AWS CLI](#)
- [Elimine un corredor con el 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 utilizar las API de administración de Kafka o Cruise Control para mover las particiones a otros agentes que desee retener 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 agente con el AWS Consola de administración

Para eliminar a los corredores con la AWS Consola de 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.

Elimine 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 [DescribeCluster](#) operación 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 el AWS API

Para eliminar a los agentes de un clúster mediante la API, consulte la referencia [UpdateBrokerCount](#) de 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. Cambiar el tamaño de sus agentes le brinda la flexibilidad de ajustar la capacidad de cómputo de su clúster de MSK en función de los cambios en sus cargas de trabajo, sin interrumpir el clúster. I/O Amazon MSK usa el mismo tamaño de agente para todos los agentes de un clúster determinado.

En el caso de agentes Standard, puede actualizar el tamaño del agente del clúster de M5 o T3 a M7g, de T3 a M5, o de M7g a M5.

 Note

No puede migrar de un tamaño de agente más grande a uno más pequeño. Por ejemplo, para. M7g.large T3.small

En el caso de los agentes Express, solo puede usar tamaños de agente M7g.

Este tema describe cómo actualizar el tamaño del agente del clúster de MSK.

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. Migrar a un tamaño de agente más grande puede aumentar el rendimiento, pero también puede implicar un mayor costo.

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 “Dimensione correctamente el clúster: número de particiones por agente Standard”](#).

Temas

- [Actualice el tamaño del bróker de clústeres de Amazon MSK mediante el Consola de administración de AWS](#)

- [Actualice el tamaño del bróker de clústeres de Amazon MSK mediante el AWS CLI](#)
- [Actualización del tamaño del agente mediante la API](#)

Actualice el tamaño del bróker de clústeres de Amazon MSK mediante el Consola de administración de AWS

Este proceso muestra cómo actualizar el tamaño del agente de clústeres de Amazon MSK mediante el Consola de administración de AWS

1. [Inicie sesión en y abra la Consola de administración de AWS consola de Amazon MSK en? https://console.aws.amazon.com/msk/home region=us-east-1#/home/](https://console.aws.amazon.com/msk/home 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.

Note

Si está activado el [reequilibrio inteligente](#) en los clústeres basados en Express recién creados, no podrá utilizar herramientas de terceros, como Cruise Control, para reequilibrar las particiones. Primero debes pausar el reequilibrio inteligente para usar la API de reasignación de particiones proporcionada por estas herramientas de terceros.

Descarga y compilación de Cruise Control

1. Cree una instancia de Amazon EC2 en la misma Amazon VPC que el clúster de Amazon MSK.
2. Instale Prometheus en la instancia de Amazon EC2 creada 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. Descargue [Cruise Control](#) en la instancia de Amazon EC2. (Si lo prefiere, también puede utilizar una instancia de Amazon EC2 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 [GetBootstrapBrokers](#) y [DescribeCluster](#) API 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.Prometheus

# 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 agentes Express, recomendamos no utilizar `DiskCapacityGoal` en ninguno de los objetivos configurados en las [configuraciones del analizador](#).

2. Edite el `config/capacityCores.json` archivo para especificar el tamaño de disco, los núcleos de CPU y in/out los límites de red correctos. En el caso de los agentes Express, la entrada de capacidad DISK solo es necesaria para configurar Cruise Control. Dado que Amazon MSK administra todo el almacenamiento de los agentes Express, debe establecer este valor en un número extremadamente alto, como `Integer.MAX_VALUE` (2147483647). 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 la CPU y in/out los límites de la red, consulte [Tipos de instancias de Amazon EC2](#).

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

Important

Solo las versiones de Cruise Control 2.5.60 o posteriores son compatibles con agentes Express, ya que los agentes Express no exponen puntos de conexión 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 instancia de Amazon EC2 con Cruise Control y Prometheus preconfigurados.
- Compatibilidad con el clúster aprovisionado de Amazon MSK.

- [Autenticación flexible con PlainText un IAM.](#)
- 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 la partición

Guía para la reasignación de particiones de Kafka

La reasignación de particiones en Kafka puede requerir un uso intensivo de recursos, ya que implica transferir grandes volúmenes de datos entre agentes, lo que puede provocar congestión de red y afectar las operaciones de los clientes. Las siguientes prácticas recomendadas ayudan a administrar la reasignación de particiones de forma eficaz mediante el ajuste de las tasas de limitación controlada, el aprovechamiento de los controles de simultaneidad y la comprensión de los tipos de reasignación, con el fin de minimizar la interrupción de las operaciones del clúster.

Note

Si tiene un clúster basado en Express recién creado, utilice el [reequilibrio inteligente](#) para la distribución automática de las particiones a medida que amplíe o reduzca los clústeres.

Administración de la simultaneidad en Cruise Control

Cruise Control proporciona parámetros de ajuste automático para controlar la simultaneidad de los movimientos de particiones y de liderazgo. Los siguientes parámetros ayudan a mantener una carga aceptable durante las reasignaciones:

- Movimientos máximos de particiones simultáneos: defina `num.concurrent.partition.movements.per.broker` para limitar la cantidad de movimientos de particiones entre agentes que se ejecutan de forma simultánea, lo que evita una utilización excesiva de la red.

Example Ejemplo

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

Esta configuración limita a cada agente a mover no más de 10 particiones en un momento dado, lo que permite equilibrar la carga entre los agentes.

Uso de la limitación controlada para controlar el ancho de banda

- **Parámetro de limitación controlada:** al realizar una reasignación de particiones con `kafka-reassign-partitions.sh`, utilice `--throttle parameter` para establecer una tasa máxima de transferencia (en bytes por segundo) para el movimiento de datos entre agentes.

Example Ejemplo

```
--throttle 5000000
```

Esto establece un ancho de banda máximo de 5 MB/s.

- **Equilibrio de la configuración de limitación controlada:** la elección de una tasa de limitación controlada adecuada es fundamental:

Si se configura un valor demasiado bajo, la reasignación puede tardar considerablemente más.

Si se configura un valor demasiado alto, los clientes pueden experimentar aumentos de latencia.

- Comience con una tasa de limitación controlada conservadora y ajústela en función de la supervisión del rendimiento del clúster. Pruebe la limitación controlada seleccionada antes de aplicarla en un entorno de producción para encontrar el equilibrio óptimo.

Prueba y validación en un entorno de preproducción

Antes de implementar reasignaciones en producción, realice pruebas de carga en un entorno de preproducción con configuraciones similares. Esto permite refinar con mayor precisión los parámetros y minimizar impactos inesperados en la producción activa.

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 “Dimensione correctamente el clúster: número de particiones por agente Standard”](#). 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 “Configuraciones de agentes”](#).

Temas

- [Disponibilidad del agente durante las actualizaciones de configuración](#)
- [Actualización de la configuración de un clúster mediante la AWS CLI](#)
- [Actualización de la configuración de un clúster de Amazon MSK mediante la API](#)

Disponibilidad del agente durante las actualizaciones de configuración

Amazon MSK mantiene una alta disponibilidad durante la mayoría de las actualizaciones de configuración del clúster. Amazon MSK realiza una actualización gradual en la que actualiza un agente a la vez. Durante este proceso, el clúster permanece disponible, aunque los agentes individuales se reinician a medida que se actualizan sus configuraciones. Sin embargo, algunos cambios de configuración pueden requerir que todos los agentes se actualicen de forma simultánea, lo que puede provocar una breve interrupción en el clúster. Para obtener más información sobre el impacto en la disponibilidad de los agentes durante las actualizaciones, consulte [Configuración de Amazon MSK aprovisionado](#).

Antes de actualizar clústeres de producción, recomendamos probar los cambios de configuración en un entorno que no sea de producción y programar las actualizaciones durante los periodos de mantenimiento.

Si experimenta algún problema al actualizar el clúster de MSK, consulte [¿Cómo soluciono los problemas al actualizar mi clúster de Amazon MSK?](#)

Actualización de la configuración de un clúster mediante la 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 [DescribeCluster](#) operación o el comando [AWS CLI describe-cluster](#). Un ejemplo de ID de versión es `KTVDPKIKX0DER`.

```
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](#).

Configurar el tipo de red de doble pila para un clúster de Amazon MSK

Amazon MSK admite el tipo de red de doble pila para los clústeres aprovisionados de MSK existentes que utilizan la versión 3.6.0 o posterior de Kafka sin coste adicional. Con las redes de doble pila, sus clústeres pueden utilizar ambas direcciones. IPv4 IPv6 Los puntos finales de doble pila también son compatibles, IPv4 por lo que mantienen la compatibilidad con versiones anteriores. Amazon MSK proporciona IPv6 soporte a través del tipo de red de doble pila, no solo como IPv6.

De forma predeterminada, los clientes se conectan a los clústeres de Amazon MSK mediante el tipo IPv4 de red. Todos los clústeres nuevos que cree también se utilizan de forma IPv4 predeterminada. Para actualizar el tipo de red de un clúster a uno de doble pila, asegúrese de cumplir los requisitos previos que se describen en la siguiente sección. A continuación, utilice la [UpdateConnectivity](#) API para actualizar la conectividad a la doble pila.

Cuando habilite las redes de doble pila en tu clúster, recibirás dos tipos de cadenas de arranque: una para la conectividad IPv4 y otra para la conectividad. IPv6 Si bien el propio clúster es compatible con ambas IPv4 y de IPv6 forma simultánea (doble pila), cada cadena de arranque es específica del protocolo: tendrás que usar cadenas de arranque para las conexiones y cadenas de IPv4 arranque para las conexiones. IPv4 IPv6 IPv6 No hay una cadena de arranque única que admita ambos protocolos. Las cadenas de IPv4 arranque existentes seguirán funcionando como antes, mientras que las nuevas cadenas de IPv6 arranque se identificarán con el prefijo «bi6-». Tenga en cuenta que la IPv6 conectividad solo está disponible para los nodos intermediarios; solo se puede acceder a ZooKeeper los nodos mediante cadenas de arranque. IPv4 Asegúrese de configurar los puertos adecuados para el protocolo que va a utilizar, ya que IPv4 las IPv6 conexiones utilizan puertos diferentes. Para obtener más información sobre los puertos necesarios, consulte [Información de puertos](#).

Todos los clústeres aprovisionados por MSK existentes que admiten redes de doble pila utilizarán las mismas cadenas de IPv6 arranque tanto para la conectividad predeterminada como para la de acceso público. Si tiene habilitado el acceso público en su clúster, las cadenas de IPv6 arranque tendrán automáticamente la capacidad de acceso público. Tenga en cuenta que si el acceso público no está habilitado en su clúster, estas cadenas de IPv6 arranque no tendrán capacidad de acceso público. Las cadenas de IPv4 arranque existentes seguirán funcionando como antes en lo que respecta a la conectividad. IPv4

 Note

Una vez que actualice el clúster para usar el tipo de red de doble pila, no podrá volver a cambiarlo al tipo de IPv4 red.

Temas

- [Requisitos previos para usar un tipo de red de doble pila](#)
- [Permisos de IAM para configurar el tipo de red de doble pila](#)
- [Configure el tipo de red de doble pila para un clúster existente](#)
- [Consideraciones sobre el uso de un tipo de red de doble pila](#)

Requisitos previos para usar un tipo de red de doble pila

Antes de configurar el tipo de red de doble pila para los clústeres, asegúrese de tener lo siguiente:

- Kafka, versión 3.6.0 o posterior, para los clústeres aprovisionados de MSK existentes.
- Todas las subredes que proporcione durante la creación del clúster deben tener asignados bloques IPv6 CIDR IPv4 y ser compatibles con el tipo de red de doble pila. Si ni siquiera una subred de tu clúster tiene bloques IPv6 CIDR asignados, no podrás usar el tipo de red de doble pila para tu clúster.
- Los tipos de instancias kafka.t3.small no admiten la conectividad de doble pila. Si tienes este tipo de instancia y quieres usar la conectividad de doble pila, primero debes actualizar a otro tipo de instancia compatible.
- Los puertos necesarios deben estar abiertos para la IPv6 conectividad. Para obtener información sobre los puertos necesarios, consulte [Información de puertos](#).
- La conectividad de doble pila no es compatible con los ZooKeeper nodos. Solo puede conectarse a ZooKeeper los nodos mediante IPv4.

Permisos de IAM para configurar el tipo de red de doble pila

Debe tener los siguientes permisos de IAM:

- `ec2:CreateTags`
- `ec2:DescribeSubnets`

- `kafka:UpdateConnectivity`

Para obtener una lista completa de los permisos necesarios para realizar todas las acciones de Amazon MSK, consulte la política AWS gestionada: [Amazon MSKFull Access](#).

Configure el tipo de red de doble pila para un clúster existente

Puede actualizar el tipo de red de un clúster aprovisionado de MSK existente mediante el Consola de administración de AWS, AWS CLI, o el SDK. AWS

Using Consola de administración de AWS

1. ¿Abrir la consola Amazon MSK en <https://console.aws.amazon.com/msk/casa?region=us-east-1#/home/>.
2. Elija el clúster aprovisionado de MSK para el que desee configurar el tipo de red de doble pila.
3. En la página de detalles del clúster, elija Propiedades.
4. En Configuración de red, elija Editar. A continuación, elija Modificar el tipo de red.

Note

Si el clúster no utiliza la versión 3.6.0 o posterior de Kafka y sus subredes no admiten la configuración de doble pila, la opción de modificar el tipo de red no estará disponible. Para cambiar el tipo de red, actualiza tu versión de Kafka y usa subredes que admitan la configuración de doble pila.

5. En Tipo de red, selecciona Doble pila.
6. Seleccione Save changes (Guardar cambios).

Using AWS CLI

Puedes usar la API de [actualización y conectividad para](#) actualizar el tipo de red de tu clúster aprovisionado de MSK existente a una pila doble. En el siguiente ejemplo, se usa el `update-connectivity` comando para establecer el tipo de red del clúster en doble pila.

En el siguiente ejemplo, sustituya el ARN del clúster de muestra, `arn:aws:kafka:us-east-1::cluster//`, por el ARN del clúster de MSK `123456789012` actual. `myCluster`

12345678-1234-1234-1234-123456789012 -1 [Para obtener la versión actual del clúster, usa el comando describe-cluster.](#)

```
aws kafka update-connectivity \
  --cluster-arn "arn:aws:kafka:us-
east-1:123456789012:cluster/myCluster/12345678-1234-1234-1234-123456789012-1" \
  --current-version "KTVPDKIKX0DER" \
  --connectivity-info '{
    "networkType": "DUAL"
  }
```

Using AWS SDK

En el siguiente ejemplo, se usa la [UpdateConnectivity](#) API para establecer el tipo de red del clúster en doble pila.

En el siguiente ejemplo, sustituya el ARN del clúster de muestra, `arn:aws:kafka:us-east-1::cluster//`, por el ARN del clúster de MSK **123456789012** actual. **myCluster 12345678-1234-1234-1234-123456789012 -1** Para obtener la versión actual del [DescribeCluster](#) clúster, usa la API.

```
import boto3

client = boto3.client("kafka")

response = client.update_connectivity(
    ClusterArn="arn:aws:kafka:us-
east-1:123456789012:cluster/myCluster/12345678-1234-1234-1234-123456789012-1",
    CurrentVersion="KTVPDKIKX0DER",
    ConnectivityInfo={
        "NetworkType": "DUAL"
    }
)

print("Connectivity update initiated:", response)
```

Consideraciones sobre el uso de un tipo de red de doble pila

- IPv6 Actualmente, el soporte solo está disponible en el modo de doble pila (IPv4 + IPv6), no solo como IPv6 -.

- Para utilizar un tipo de red de doble pila para los clústeres aprovisionados de Amazon MSK, se necesita la versión 3.6.0 o posterior de Kafka.
- El tipo de red de doble pila no está disponible para la conectividad privada de varias VPC.
- IPv6 la conectividad para los nodos de Zookeeper no está disponible. Solo puede conectarse a los ZooKeeper nodos que utilicen. IPv4
- Puede cambiar el tipo de red de IPv4 doble pila para un clúster existente solo si todas sus subredes admiten el tipo de red de doble pila.
- No puedes volver al tipo de IPv4 red después de habilitar la doble pila. Para volver a la versión anterior, debe eliminar y volver a crear el clúster.
- Debe tener los siguientes permisos de IAM:
 - `ec2:CreateTags`, `ec2:DescribeSubnets` y `kafka:UpdateConnectivity`

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 Consola de administración de AWS

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

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 el 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 de cargas de trabajo de Kafka a un clúster de Amazon MSK

Amazon MSK Replicator admite la replicación entre clústeres de Amazon MSK del mismo. Cuenta de AWS También admite migraciones entre clústeres de Apache Kafka que no son de MSK y clústeres de Amazon MSK. Para la replicación entre cuentas entre clústeres de Amazon MSK, debe usar Apache MirrorMaker 2.0.

[Replicador MSK](#) es una solución completamente administrada y sin servidor que automatiza la migración de datos a Amazon MSK. Replicador MSK gestiona el escalado, la supervisión y

las tareas de mantenimiento sin necesidad de administrar la infraestructura. También mantiene las configuraciones de temas y los desplazamientos de los grupos de consumidores durante la migración, y se integra con otros Servicios de AWS.

[Apache MirrorMaker 2.0](#) es una herramienta de código abierto que requiere configuración y administración manuales, pero proporciona un control detallado del proceso de migración. Puede definir reglas de replicación personalizadas y migrar entre cualquier clúster de Apache Kafka, independientemente de la plataforma de alojamiento o de una plataforma a otra. Cuentas de AWS Para obtener información sobre cómo MirrorMaker migrar sus clústeres, consulte [Geo-Replication \(Cross-Cluster Data Mirroring\)](#). Recomendamos configurarlo MirrorMaker en una configuración de alta disponibilidad.

Para obtener más información sobre la migración a Amazon MSK con MSK Replicator, consulte [Migración de clústeres de Apache Kafka autogestionados y de terceros a agentes de Amazon MSK Express con Amazon MSK Replicator](#).

Eliminación de un clúster de Amazon MSK aprovisionado

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 el Consola de administración de AWS](#)
- [Elimine un clúster aprovisionado de Amazon MSK mediante el AWS CLI](#)
- [Eliminación de un clúster de Amazon MSK aprovisionado mediante la API](#)

Elimine un clúster aprovisionado de Amazon MSK mediante el Consola de administración de AWS

Este proceso describe cómo eliminar un clúster de Amazon MSK aprovisionado mediante la Consola de administración de AWS. 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. [Inicie sesión en y abra la Consola de administración de AWS consola de Amazon MSK en?](https://console.aws.amazon.com/msk/home 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 el AWS CLI

Este proceso describe cómo eliminar un clúster de MSK aprovisionado mediante la 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
```

Eliminación de un clúster de Amazon MSK aprovisionado mediante la API

La API de Amazon MSK permite crear y administrar mediante programación el clúster de MSK aprovisionado como parte de procesos automatizados de aprovisionamiento de infraestructura o de scripts de implementación. Este proceso describe cómo eliminar un clúster de Amazon MSK aprovisionado 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 de Amazon MSK aprovisionado ofrecen una amplia gama de características y capacidades que ayudan a optimizar el rendimiento del clúster y a satisfacer las necesidades de streaming. Los temas que se presentan a continuación describen estas funcionalidades en detalle.

- la [Consola de administración de AWS](#),

- La [referencia de la API de Amazon MSK](#)
- La [referencia de los comandos de la CLI de Amazon MSK](#)

Temas

- [Tipos de agentes de Amazon MSK](#)
- [Tamaños del agente de Amazon MSK](#)
- [Administración del almacenamiento para agentes Standard](#)
- [Configuración de Amazon MSK aprovisionado](#)
- [Reequilibrio inteligente para clústeres](#)
- [Aplicación de parches en clústeres de MSK aprovisionados](#)
- [El agente no está en línea y el cliente realiza una conmutación por error](#)
- [Seguridad en Amazon MSK](#)
- [Registro de Amazon MSK](#)
- [Administración de metadatos](#)
- [Tema: Operaciones](#)
- [Recursos de Amazon MSK](#)
- [Versiones de Apache Kafka](#)
- [Solución de problemas del clúster de Amazon MSK](#)

Tipos de agentes de Amazon MSK

MSK aprovisionado ofrece dos tipos de agentes: Standard y Express. Los corredores estándar le brindan la mayor flexibilidad para configurar sus clústeres, mientras que los corredores Express ofrecen más elasticidad, rendimiento y resiliencia y ease-of-use para ejecutar aplicaciones de streaming de alto rendimiento.

Consulte los siguientes temas para obtener más detalles sobre cada oferta. La siguiente tabla también destaca la comparación de características clave entre los agentes Standard y Express.

Característica	Agente Standard	Agente Express
Administración del almacenamiento	Administrado por el cliente (las características incluyen	Completamente administrado por MSK

Característica	Agente Standard	Agente Express
	almacenamiento EBS, almacenamiento por niveles, rendimiento de almacenamiento aprovisionado, escalado automático y alertas de capacidad de almacenamiento)	
Instancias compatibles	T3, M5, M7g	M7g
Consideraciones de dimensionamiento y escalado	Rendimiento, conexiones, particiones, almacenamiento	Rendimiento, conexiones, particiones
Escalado de agentes	Escalado vertical y horizontal	Escalado vertical y horizontal
Versiones de Kafka	Consulte Versiones de Apache Kafka	Comienza en la versión 3.6
Configuración de Apache Kafka	Más configurable	Administrado principalmente por MSK para una mayor resiliencia
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 es posible cambiar un clúster de MSK aprovisionado de un tipo de agente Standard a un tipo de agente Express mediante la API de MSK. Debe crear un nuevo clúster con el tipo de agente deseado (Standard o Express).

Temas

- [Agentes Standard de Amazon MSK](#)
- [Agentes Express de Amazon MSK](#)

Agentes Standard de Amazon MSK

Los agentes Standard para MSK aprovisionado ofrecen la mayor flexibilidad para configurar el rendimiento del clúster. Puede elegir entre una amplia gama de configuraciones de clúster para lograr las características de disponibilidad, durabilidad, rendimiento y latencia que requieren las aplicaciones. También puede aprovisionar capacidad de almacenamiento y aumentarla según sea necesario. Amazon MSK gestiona el mantenimiento del hardware de los agentes Standard y de los recursos de almacenamiento asociados, y repara automáticamente los problemas de hardware que puedan surgir. Puede encontrar más detalles en este documento sobre diversos temas relacionados con los agentes Standard, incluidos temas sobre [administración del almacenamiento](#), [configuraciones](#) y [mantenimiento](#).

Agentes Express de Amazon MSK

Los agentes Express para MSK aprovisionado simplifican la administración de Apache Kafka, resultan más rentables para operar a escala y ofrecen mayor elasticidad 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 instancia seleccionado, cada nodo de agente puede proporcionar hasta tres veces más rendimiento por agente, escalar verticalmente hasta 20 veces más rápido y recuperarse un 90 % más rápido en comparación con los agentes Standard de Apache Kafka. Los agentes Express vienen preconfigurados con los valores predeterminados de prácticas recomendadas de Amazon MSK y aplican cuotas de rendimiento del cliente para minimizar la contención de recursos entre los clientes y las operaciones en segundo plano de Kafka.

A continuación, se presentan algunas características clave y capacidades que se deben considerar al utilizar agentes Express.

- Sin administración del almacenamiento: los agentes Express eliminan la necesidad de [aprovisionar o administrar recursos de almacenamiento](#). Obtiene un almacenamiento elástico, prácticamente ilimitado y totalmente gestionado. pay-as-you-go Para casos de uso de alto rendimiento, no es necesario analizar las interacciones entre las instancias de computación y los volúmenes de almacenamiento, ni los cuellos de botella de rendimiento asociados. Estas capacidades simplifican

la administración del clúster y eliminan la sobrecarga operativa asociada a la administración del almacenamiento.

- **Escalado más rápido:** los agentes Express permiten escalar el clúster y mover particiones hasta 20 veces más rápido que con agentes Standard. Esta capacidad es fundamental cuando necesita escalar horizontalmente el clúster para gestionar picos de carga entrantes o reducir horizontalmente para disminuir 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 reequilibrarlo](#) si desea obtener más información sobre cómo escalar su clúster.
- **Mayor rendimiento:** los agentes Express ofrecen hasta tres veces más rendimiento por agente que los agentes Standard. Por ejemplo, puede escribir datos de forma segura hasta 500 MBps con cada broker Express de 7g.16x de tamaño, en comparación con los 153,8 con el broker 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).
- **Configurado para alta resiliencia:** los agentes Express ofrecen automáticamente diversas prácticas recomendadas para mejorar la resiliencia del clúster. Estas incluyen barreras de protección en configuraciones críticas de Apache Kafka, cuotas de rendimiento y reservas de capacidad para operaciones en segundo plano y reparaciones no planificadas. Estas capacidades hacen que sea más seguro y sencillo ejecutar aplicaciones de Apache Kafka a gran escala. Consulte las secciones [Configuraciones de agentes Express](#) y [Cuota de agentes Express de Amazon MSK](#) para obtener más detalles.
- **Sin periodos de mantenimiento:** no existen periodos de mantenimiento para los agentes Express. Amazon MSK actualiza automáticamente el hardware del clúster de forma continua. Consulte [Aplicación de parches para agentes Express](#) para obtener más detalles.

Información adicional sobre los agentes Express

- Los corredores de Express trabajan con Apache Kafka APIs, pero aún no son totalmente compatibles con la API. KStreams
- Los corredores Express solo están disponibles en una AZs configuración de 3.
- Los agentes Express solo están disponibles en determinados tamaños de instancia. Consulte [Precios de Amazon MSK](#) para obtener la lista actualizada.
- Los corredores Express son compatibles con las siguientes versiones de Apache Kafka: 3.6, 3.8 y 3.9.
- Los corredores Express se pueden crear con el KRaft modo a partir de la versión 3.9 de Apache Kafka.

Consulte estos blogs

Para obtener más información sobre los agentes Express de MSK y ver un ejemplo práctico de agentes Express en uso, lea los siguientes blogs:

- [Presentación de agentes Express para Amazon MSK para ofrecer alto rendimiento y un escalado más rápido para los clústeres de Kafka](#)
- [Agentes Express para Amazon MSK: escalado de Kafka con turbo y un rendimiento hasta 20 veces más rápido](#)

Este blog muestra cómo los agentes Express:

- Ofrecen mayor rendimiento, permiten un escalado rápido y mejoran el tiempo de recuperación ante fallos
- Deshágase de las complejidades de la administración del almacenamiento

Tamaños del agente de Amazon MSK

Cuando crea un clúster de Amazon MSK aprovisionado, especifica el tamaño de los agentes que desea que tenga. Según el [tipo de agente](#), Amazon MSK admite los siguientes tamaños de agente.

Tamaños de agentes 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 de agentes Express

- 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 AWS regiones. Consulte las tablas de precios actualizadas de instancias de agentes en la [página de precios de Amazon MSK](#) para obtener la lista más reciente de instancias disponibles por región.

Otras notas sobre los tamaños de agentes

- 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 de MSK aprovisionado que ejecutan las versiones 2.8.2 y 3.3.2 y posteriores 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 [Prácticas recomendadas para agentes Standard y Express](#).

Administración del almacenamiento para agentes Standard

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

Note

Con los [agentes Express](#), no es necesario aprovisionar ni administrar ningún recurso de almacenamiento utilizado por los datos. Esto simplifica la administración del clúster y elimina una de las causas comunes de problemas operativos en los clústeres de Apache Kafka. Además, reduce los costos, ya que no es necesario aprovisionar capacidad de almacenamiento inactiva y solo se paga por lo que se utiliza.

Tipo de agente estándar

Con [agentes 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 administración del rendimiento, consulte [???](#).

Temas

- [Almacenamiento por niveles para agentes Standard](#)
- [Escalado vertical del almacenamiento de los agentes Standard de Amazon MSK](#)
- [Administración del rendimiento del almacenamiento para agentes Standard en un clúster de Amazon MSK](#)

Almacenamiento por niveles para agentes Standard

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 las API introducidas en [KIP-405: Kafka Tiered Storage](#).

Para obtener información sobre el dimensionamiento, la supervisión y la optimización del clúster de Amazon MSK con almacenamiento por niveles, consulte [Prácticas recomendadas para ejecutar cargas de trabajo de producción con almacenamiento por 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 flujos existente y las API de Kafka.
- 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 AWS GovCloud (EE. UU.).
- 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 habilitado el almacenamiento por niveles tengan configurada la política de limpieza únicamente como “ELIMINAR”.
- El clúster con almacenamiento por niveles no admite modificar la política `log.cleanup.policy` de un tema después de su creación.
- El almacenamiento por niveles se puede desactivar 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 herramienta kafka-log-dirs no puede informar sobre el tamaño de los datos del 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 predeterminada y las restricciones que debe tener en cuenta al configurar el almacenamiento por niveles en el nivel de tema, 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.

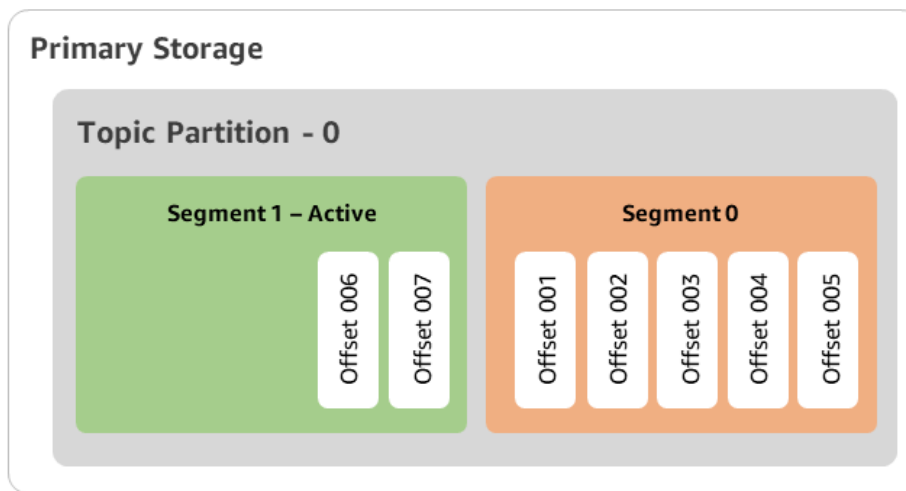
- Las configuraciones de retención habituales se aplican a todo el registro. Esto incluye sus partes por niveles y principales.
- La configuración `local.retention.ms` o `local.retention.bytes` controla la retención de los mensajes en el almacenamiento principal. Apache Kafka copia los segmentos de registros cerrados en un almacenamiento por niveles tan pronto como se cierran (en función de `segment.bytes` o `segment.ms`), independientemente de la configuración de retención local. Una vez copiados los segmentos al almacenamiento por niveles, permanecen en el almacenamiento principal hasta que se alcanzan los umbrales `local.retention.ms` o `local.retention.bytes`. En ese momento, los datos se eliminan del almacenamiento principal, pero permanecen disponibles en el almacenamiento por niveles. Esto le permite conservar los datos recientes en un almacenamiento primario de alto rendimiento para un acceso rápido, mientras que los datos más antiguos se almacenan en un almacenamiento en niveles de bajo costo.
- 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.

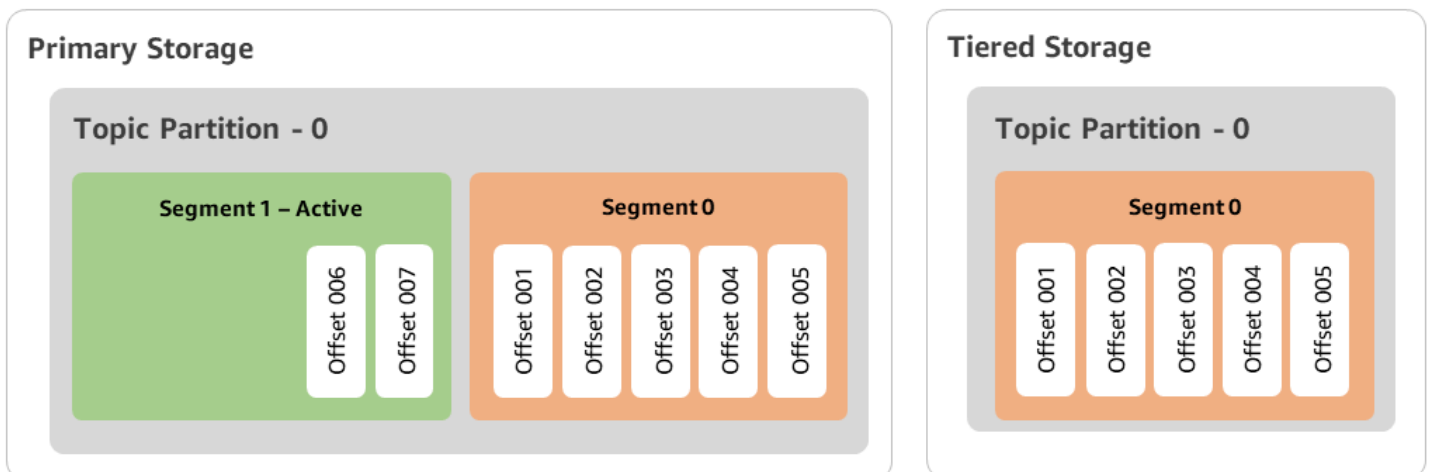
Momento T0: antes de habilitar el almacenamiento por niveles.

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.



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

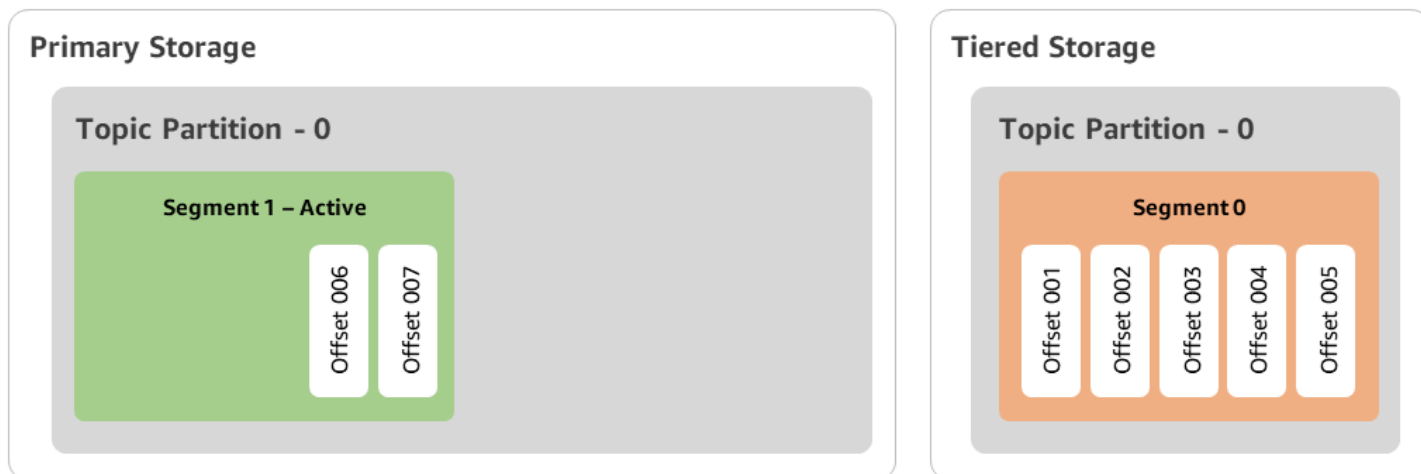
Tras activar el almacenamiento por niveles para este tema, Apache Kafka copia el segmento 0 del registro cerrado en el almacenamiento por niveles tan pronto como se cierra. El segmento se cierra según la configuración `segment.bytes` o `segment.ms`, no según la configuración de retención. Apache Kafka también conserva una copia en el almacenamiento principal. El segmento activo 1 aún no es apto para copiarse en un almacenamiento por niveles porque sigue activo y no se ha cerrado. 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`, retención. `ms/bytes`)



Momento T2: retención local vigente.

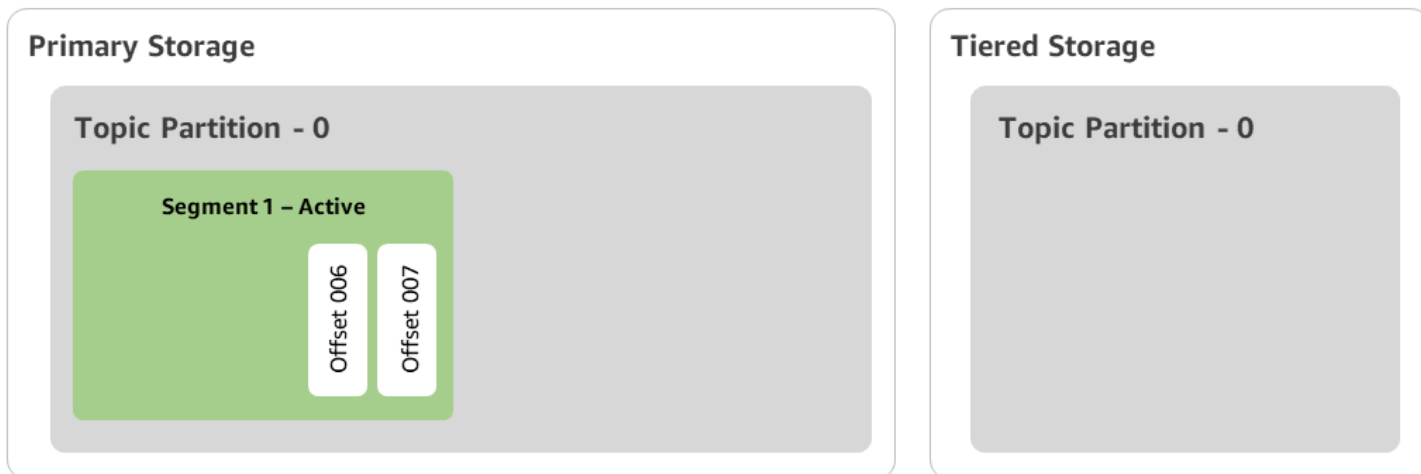
Después de 2 días, se alcanza el umbral de retención local para el segmento 0. Esto se determina si se establece `local.retention.ms` en 2 días. El segmento 0 ahora se ha eliminado del almacenamiento

principal, pero permanece disponible en el almacenamiento por niveles. Tenga en cuenta que el segmento 0 ya estaba copiado en el almacenamiento por niveles en el momento T1 cuando se cerró, no en el momento T2, cuando expiró la retención local. El segmento activo 1 aún no es apto para ser eliminado ni copiado en un almacenamiento por niveles porque sigue activo.



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 Consola de administración de AWS

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

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 el 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://broker-node-group-info.json \  
  --number-of-broker-nodes 3 \  
  --kafka-version "3.6.0" \  
  --storage-type "RAID0" \  
  --availability-zones us-east-1a,us-east-1b,us-east-1c
```

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

Cree un tema de Kafka con el almacenamiento por niveles habilitado con el 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 la configuración](#) para obtener más información sobre los atributos de configuración compatibles.

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=1000000 --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 Consola de administración de AWS](#).

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 AWS CLI

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 la configuración](#) para obtener más detalles sobre los atributos de configuración compatibles.

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 `KTVDPKIKX0DER`.

```
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 Consola de administración de AWS.

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 la configuración](#) para obtener más información sobre los atributos de configuración compatibles.

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.

Escalado vertical del almacenamiento de los agentes Standard 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 [UpdateBrokerStorage](#) operación. Para obtener información sobre el tamaño correcto del almacenamiento, consulte [the section called “Prácticas recomendadas para agentes Standard”](#).

Puede utilizar el almacenamiento por niveles para escalar verticalmente a cantidades ilimitadas de almacenamiento para el agente. Consulte, [Almacenamiento por niveles para agentes Standard](#).

Temas

- [Escalado automático para clústeres de Amazon MSK](#)
- [Escalado manual para agentes Standard](#)

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 [Dimensione correctamente el clúster: número de agentes Standard 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 MSK](#).
- Amazon MSK no admite el escalado automático en las regiones Asia-Pacífico (Osaka), África (Ciudad del Cabo) y Asia-Pacífico (Malasia).
- 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 elimina 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 acerca del seguimiento de objetivos, consulte [Target tracking scaling policies for 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 [ListClusterOperations](#) operación.

Configuración del escalado automático para su clúster de Amazon MSK

Puede utilizar la consola de Amazon MSK, la API de Amazon MSK o 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

- [Configure el escalado automático con Amazon MSK Consola de administración de AWS](#)
- [Configuración del escalado automático mediante la CLI](#)
- [Configuración del escalado automático de Amazon MSK con la API](#)

Configure el escalado automático con Amazon MSK Consola de administración de AWS

En este proceso, se describe cómo utilizar la consola de Amazon MSK para implementar el escalado automático para almacenamiento.

1. ¿Iniciar sesión en la Consola de administración de AWS 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.

Configuración del 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 comando para registrar un objetivo de utilización del almacenamiento [RegisterScalableTarget](#).
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 agentes Standard

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 información de los corredores mediante Consola de administración de AWS

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 corredores 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 [DescribeCluster](#) operación o el comando [AWS CLI describe-cluster](#). Un ejemplo de ID de versión es KTVDPKIKX0DER.

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](#).

Administración del rendimiento del almacenamiento para agentes Standard en un clúster de Amazon MSK

Para obtener información sobre cómo aprovisionar el rendimiento mediante la consola de Amazon MSK, la CLI y la API, 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](#)
- [Aprovisionamiento del rendimiento de almacenamiento para agentes Standard 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 atascos en el rendimiento de los agentes: el rendimiento del volumen, el rendimiento de la red de Amazon EC2 a 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 agentes pueden deberse al rendimiento de la red de Amazon EC2 a Amazon EBS y al rendimiento de salida de Amazon EC2.

El rendimiento de salida de Amazon EC2 se ve afectado por la cantidad de grupos de consumidores y de consumidores por grupos de consumidores. Además, tanto el rendimiento de la red de Amazon

EC2 a Amazon EBS como el rendimiento de salida de Amazon EC2 son más altos para los tamaños del agente más grandes.

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/second
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.

$$\text{Average storage throughput in MiB/s} = (\text{Sum}(\text{VolumeReadBytes}) + \text{Sum}(\text{VolumeWriteBytes})) / (60 * 1024 * 1024)$$

Para obtener información sobre las métricas `VolumeReadBytes` y `VolumeWriteBytes`, consulte [the section called “PER_BROKER: Monitorización de niveles”](#).

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.

Aprovisionamiento del rendimiento de almacenamiento para agentes Standard 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 Consola de administración de AWS](#)
- [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 Consola de administración de AWS

Este proceso muestra un ejemplo de cómo puede usarlo Consola de administración de AWS para crear un clúster de Amazon MSK con el rendimiento aprovisionado habilitado.

1. [Inicie sesión en y abra la Consola de administración de AWS consola de Amazon MSK en?](https://console.aws.amazon.com/msk/home?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 ID de subred y de grupo de seguridad por 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. Ejecuta el siguiente AWS CLI comando desde el directorio en el que guardaste 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, usa CreateCluster la V2.](#)

Configuración de Amazon MSK aprovisionado

Amazon MSK proporciona configuraciones predeterminadas para los agentes, 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 el clúster, existe un conjunto distinto de configuraciones predeterminadas y un conjunto diferente de configuraciones que puede modificar. Consulte las secciones siguientes para obtener más detalles sobre cómo configurar agentes Standard y Express.

Temas

- [Configuraciones de agentes Standard](#)
- [Configuraciones de agentes Express](#)
- [Operaciones de configuración de agentes](#)

Configuraciones de agentes Standard

Esta sección describe las propiedades de configuración de los agentes Standard.

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

Puede usar Amazon MSK para crear una configuración personalizada de MSK en la que establezca las siguientes propiedades de configuración de Apache Kafka. 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](#).

Name	Description (Descripción)
<code>allow.everyone.if.no.acl.found</code>	Si desea establecer esta propiedad en <code>false</code> , primero, asegúrese de definir las ACL de Apache Kafka para su clúster. Si establece esta propiedad en <code>false</code> y no define primero las ACL de Apache Kafka, perderá el acceso al clúster. Si eso ocurre, puede volver a actualizar la configuración y establecer esta propiedad en <code>true</code> para recuperar el acceso al clúster.
<code>auto.create.topics.enable</code>	Habilita la creación automática de temas en el servidor.
<code>compression.type</code>	El tipo de compresión final para un tema determinado. Puede establecer esta propiedad en los códecs de compresión estándar (<code>gzip</code> , <code>snappy</code> , <code>lz4</code> y <code>zstd</code>). Además, acepta <code>uncompressed</code> . Este valor equivale a la ausencia de compresión. Si establece el valor en <code>producer</code> , se retendrá el códec de compresión original que configuró el productor.

Name	Description (Descripción)
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 administración.
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.

Name	Description (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 desencadena 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 frecuentes, pero también implica un gasto de espacio superior en el registro.

Name	Description (Descripción)
<code>log.cleanup.policy</code>	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 <code>delete</code> y <code>compact</code> . En el caso de los clústeres habilitados para el almacenamiento por niveles, la única política válida es <code>delete</code> .
<code>log.flush.interval.messages</code>	Número de mensajes acumulados en una partición de registro antes de que se vacíen en el disco.
<code>log.flush.interval.ms</code>	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 <code>log.flush.scheduler.interval.ms</code> . El valor mínimo es 0.
<code>log.message.timestamp.difference.max.ms</code>	Esta configuración quedó obsoleta en Kafka 3.6.0. Se agregaron dos configuraciones, <code>log.message.timestamp.before.max.ms</code> y <code>log.message.timestamp.after.max.ms</code> . 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 <code>log.message.timestamp.type=CreateTime</code> , se rechaza un mensaje si la diferencia en la marca temporal supera este umbral. Esta configuración se ignora si es <code>LogAppendTime</code> <code>log.message.timestamp.type=</code> .

Name	Description (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 <code>CreateTime</code> y <code>LogAppendTime</code> .
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 <code>log.retention.ms</code> .
log.retention.minutes	Número de minutos que se conserva un archivo de registro antes de eliminarlo, cantidad secundaria de la propiedad <code>log.retention.ms</code> . Si no establece este valor, se utiliza el valor de <code>log.retention.hours</code> .
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 <code>log.retention.minutes</code> .
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 <code>log.roll.hours</code> . 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.

Name	Description (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 <code>max.message.bytes</code>.
min.insync.replicas	Cuando un productor establece las confirmaciones en <code>"all"</code> (o <code>"-1"</code>), el valor de <code>min.insync.replicas</code> 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). <code>NotEnoughReplicas</code> <code>NotEnoughReplicasAfterAppend</code> Puede utilizar valores de <code>min.insync.replicas</code> y confirmaciones para reforzar las garantías de durabilidad. Por ejemplo, puede crear un tema con un factor de replicación de 3, establecer <code>min.insync.replicas</code> en 2 y producirlo con las confirmaciones de <code>"all"</code>. Esto garantiza que el productor emita una excepción si la mayoría de las réplicas no reciben una escritura.

Name	Description (Descripción)
num.io.threads	El número de subprocesos que el servidor utiliza para procesar las solicitudes, que puede incluir un disco I/O.
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 corredor 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 consumidores independientes (es decir, los que utilizan asignación manual), las compensaciones vencen después de la hora de la última confirmación más este periodo de retención.

Name	Description (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.message.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 propiedades 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.

Name	Description (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 milisegundos, 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 <code>org.apache.kafka.common.replica.RackAwareReplicaSelector</code>. 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.

Name	Description (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 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.
transaction.state.log.min.isr	Se ha anulado la configuración min.insync.replicas para el tema de la transacción.
transaction.state.log.replication.factor	El factor de replicación del tema de transacción. 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.

Name	Description (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 transacción del coordinador. Esta configuración también influye en el vencimiento del ID del productor , ya que hace que los ID de productor venzan cuando este tiempo transcurra después de la última escritura con el ID de productor indicado. Los ID de productor pueden vencer antes si se elimina la última escritura del ID del 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 Le recomendamos que establezca este valor en 10.000 para evitar el tiempo T3.small de inactividad del clúster.

Name	Description (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 agentes”](#). 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`.

Topic-level Configuración 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 sobre las propiedades de configuración a nivel de tema y ejemplos sobre cómo configurarlas, consulte las [Topic-Level configuraciones](#) 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](#).

Name	Description (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 o, el recurso no tiene ACL asociadas. En este caso, si establece esta propiedad como <code>true</code> , todos los usuarios pueden acceder al recurso, no solo los superusuarios.	<code>true</code>	<code>true</code>
auto.create.topics.enable	Habilita la creación automática de un tema en el servidor.	<code>false</code>	<code>false</code>

Name	Description (Descripción)	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
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.

Name	Description (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 retención. ms/ bytes ajuste 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

Name	Description (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]		

Name	Description (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 local 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 retención. ms/bytes ajuste 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 permanecer los	-2 para un número ilimitado	-2 para un número ilimitado

Name	Description (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. Los valores válidos son números enteros mayores que 0.		

Name	Description (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 quedó obsoleta en Kafka 3.6.0. Se agregaron 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 2.8.2 con funcionalidad por niveles y Kafka 3.7.x con funcionalidad por niveles.

Name	Description (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

Name	Description (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 NotEnoughReplicas opciones). NotEnoughReplicasAfterAppend Cuando utiliza de forma conjunta los valores de min.insync.replicas y las confirmaciones, puede aplicar unas garantías de durabilidad	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.

Name	Description (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 el servidor utiliza para generar solicitudes, que puede incluir un disco I/O.	8	max(8, vCPUs), donde las vCPU dependen del tamaño de la instancia del agente
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 las vCPU dependen del tamaño de la instancia del agente

Name	Description (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 aumenta este valor, puede aumentar el grado de I/O paralelo en el agente seguidor.	2	$\max(2, \text{vCPUs} / 4)$, donde las vCPU dependen del tamaño de la instancia del agente
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

Name	Description (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	$\max(10, \text{vCPUs} * 0.67)$, donde las vCPU dependen del tamaño de la instancia del agente

Name	Description (Descripción)	Valor predeterminado para el clúster de almacenamiento sin niveles	Valor predeterminado para el clúster habilitado para el almacenamiento por niveles
remote.storage.enabled	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

Name	Description (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

Name	Description (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 transfieren 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.

Name	Description (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

Name	Description (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 de la sesión de Apache en milisegundos. ZooKeeper	18000	18000
zookeeper.set.acl	El cliente configurado para que utilice ACL seguras.	false	false

Para obtener información sobre cómo especificar 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` no puede igualar ni superar la retención. `ms/bytes`. Esta es la configuración de retención del almacenamiento por niveles.
- El valor predeterminado para `local.retention.ms/bytes` es -2. Esto significa que el valor `retention.ms` se usa para `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 retención. `ms/bytes` es -1. Esto significa una retención infinita.

- El valor mínimo de `local.retention.ms/bytes` es -2. Esto significa una retención infinita para el almacenamiento local. Coincide con la retención `ms/bytes` estableciendo 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 del almacenamiento por niveles, consulte [Restricciones y limitaciones de almacenamiento por niveles para clústeres de Amazon MSK](#).

Configuraciones de agentes Express

Apache Kafka tiene cientos de configuraciones de agente que puede usar para ajustar el rendimiento del clúster de Amazon MSK aprovisionado. Establecer valores erróneos o subóptimos puede afectar la fiabilidad y el rendimiento del clúster. Los agentes Express mejoran la disponibilidad y la durabilidad de los clústeres de Amazon MSK aprovisionados al establecer valores óptimos para configuraciones críticas y protegerlos frente a errores de configuración comunes. Existen tres categorías de configuraciones basadas en el acceso de lectura y escritura: [read/write \(editables\)](#), de [solo lectura](#) y no configurables. Algunas configuraciones aún usan el valor predeterminado de Apache Kafka correspondiente a la versión de Apache Kafka en la que se ejecuta el clúster. Marcamos estas configuraciones como Valores predeterminados de Apache Kafka.

Temas

- [Configuraciones de broker MSK Express personalizadas \(acceso\) Read/Write](#)
- [Configuraciones de solo lectura para agentes Express](#)

Configuraciones de broker MSK Express personalizadas (acceso) Read/Write

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 agentes de Apache Kafka pueden ser estáticas o dinámicas. Las configuraciones estáticas requieren un reinicio del agente para que los cambios surtan efecto, mientras que las configuraciones dinámicas no requieren el reinicio del agente. Para obtener más información sobre las propiedades de configuración y los modos de actualización, consulte [Updating broker configs](#).

Temas

- [Configuraciones estáticas en agentes Express de MSK](#)
- [Configuraciones dinámicas en agentes Express](#)

- [Topic-level configuraciones en Express Brokers](#)

Configuraciones estáticas en agentes Express de MSK

Puede utilizar Amazon MSK para crear un archivo de configuración personalizado de MSK en el que defina las siguientes propiedades estáticas. Amazon MSK establece y administra automáticamente todas las demás propiedades que no configure. Puede crear y actualizar archivos de configuración estática desde la consola de MSK o mediante el [comando de configuraciones](#).

Propiedad	Description (Descripción)	Valor predeterminado
<code>allow.everyone.if.no.acl.found</code>	Si desea establecer esta propiedad en falsa, primero asegúrese de definir listas de control de acceso (ACL) de Apache Kafka para el clúster. Si establece esta propiedad en falso y no define primero las ACL de Apache Kafka, perderá el acceso al clúster. Si eso ocurre, puede actualizar nuevamente la configuración y establecer esta propiedad en verdadero para recuperar el acceso al clúster.	true
<code>auto.create.topics.enable</code>	Habilita la creación automática de un tema en el servidor.	false
<code>compression.type</code>	Especifique el tipo de compresión final para un tema determinado. Esta configuración admite los códecs de compresión estándar: gzip, snappy, lz4 y zstd. Esta configuración también acepta <code>uncompressed</code> ,	Valor predeterminado de Apache Kafka

Propiedad	Description (Descripción)	Valor predeterminado
	que es equivalente a no usar compresión, y producer, que significa conservar el códec de compresión original establecido por 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.	Valor predeterminado de Apache Kafka
delete.topic.enable	Habilita la operación para eliminar un tema. Si desactiva esta configuración, no podrá eliminar un tema usando la herramienta de administración.	Valor predeterminado de Apache Kafka
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.	Valor predeterminado de Apache Kafka

Propiedad	Description (Descripción)	Valor predeterminado
<code>group.max.session.timeout.ms</code>	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.	Valor predeterminado de Apache Kafka
<code>leader.imbalance.per.broker.percentage</code>	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.	Valor predeterminado de Apache Kafka
<code>log.cleanup.policy</code>	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 <code>delete</code> y <code>compact</code> . Para clústeres con almacenamiento por niveles habilitado, la política válida es únicamente <code>delete</code> .	Valor predeterminado de Apache Kafka

Propiedad	Description (Descripción)	Valor predeterminado
<code>log.message.timestamp.after.max.ms</code>	La diferencia de marca de tiempo permitida entre la marca de tiempo del mensaje y la marca de tiempo del agente. La marca de tiempo del mensaje puede ser posterior o igual a la marca de tiempo del agente, con la diferencia máxima permitida determinada por el valor establecido en esta configuración. Si <code>log.message.timestamp.type=CreateTime</code>, el mensaje se rechazará si la diferencia entre las marcas de tiempo supera este umbral especificado. Esta configuración se ignora si <code>log.message.timestamp.type=LogAppendTime</code>.	86400000 (24 * 60 * 60 * 1000 ms, es decir, 1 día)

Propiedad	Description (Descripción)	Valor predeterminado
<code>log.message.timestamp.before.max.ms</code>	La diferencia de marca de tiempo permitida entre la marca de tiempo del agente y la marca de tiempo del mensaje. La marca de tiempo del mensaje puede ser anterior o igual a la marca de tiempo del agente, con la diferencia máxima permitida determinada por el valor establecido en esta configuración. Si <code>log.message.timestamp.type=CreateTime</code>, el mensaje se rechazará si la diferencia entre las marcas de tiempo supera este umbral especificado. Esta configuración se ignora si <code>log.message.timestamp.type=LogAppendTime</code>.	86400000 (24 * 60 * 60 * 1000 ms, es decir, 1 día)
<code>log.message.timestamp.type</code>	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 <code>CreateTime</code> y <code>LogAppendTime</code> .	Valor predeterminado de Apache Kafka
<code>log.retention.bytes</code>	Tamaño máximo del registro antes de eliminarlo.	Valor predeterminado de Apache Kafka

Propiedad	Description (Descripción)	Valor predeterminado
log.retention.ms	Cantidad de milisegundos durante los cuales se conserva un archivo de registro antes de eliminarlo.	Valor predeterminado de Apache Kafka
max.connections.per.ip	El número máximo de conexiones permitidas desde cada dirección IP. Este valor se puede establecer en 0 si existen sobrescrituras configuradas mediante la propiedad <code>max.connections.per.ip.overrides</code> . Las nuevas conexiones procedentes de la dirección IP se descartan cuando se alcanza el límite.	Valor predeterminado de Apache Kafka
max.incremental.fetch.session.cache.slots	Número máximo de sesiones de recuperación incrementales que se conservan.	Valor predeterminado de Apache Kafka

Propiedad	Description (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 mediante la configuración <code>max.message.bytes</code> del tema.	Valor predeterminado de Apache Kafka
num.partitions	Número predeterminado de particiones por tema.	1

Propiedad	Description (Descripción)	Valor predeterminado
<code>offsets.retention.minutes</code>	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 usan asignación manual), los desplazamientos expiran después del tiempo transcurrido desde la última confirmación más este período de retención.	Valor predeterminado de Apache Kafka
<code>replica.fetch.max.bytes</code>	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 <code>message.max.bytes</code> (configuración del agente) o <code>max.message.bytes</code> (configuración del tema) definen el tamaño de lote de registro máximo que acepta el agente.	Valor predeterminado de Apache Kafka

Propiedad	Description (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 desea permitir que los consumidores obtengan datos de la réplica más cercana, establezca esta propiedad en <code>org.apache.kafka.common.replica.RackAwareReplicaSelector</code> .	Valor predeterminado de Apache Kafka
socket.receive.buffer.bytes	El búfer <code>SO_RCVBUF</code> 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 <code>SO_SNDBUF</code> de los sockets del servidor de sockets. Si el valor es -1, se utiliza el sistema operativo predeterminado.	102400

Propiedad	Description (Descripción)	Valor predeterminado
transaction.max.timeout.ms	Tiempo de espera máximo para las transacciones. Si el tiempo de transacción solicitado o 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.	Valor predeterminado de Apache Kafka

Propiedad	Description (Descripción)	Valor predeterminado
<code>transactional.id.expiration.ms</code>	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 expiración del identificador del productor, ya que provoca que los identificadores de productor expiren cuando transcurre este tiempo desde la última escritura realizada con el identificador de productor correspondiente. Los ID de productor pueden vencer antes si se elimina la última escritura del ID del productor debido a la configuración de retención del tema. El valor mínimo de esta propiedad es de 1 milisegundo.	Valor predeterminado de Apache Kafka

Configuraciones dinámicas en agentes Express

Puede utilizar la AlterConfig API de Apache Kafka o la `Kafka-configs.sh` herramienta para editar las siguientes configuraciones dinámicas. Amazon MSK establece y administra automáticamente todas las demás propiedades que no configure. Puede establecer dinámicamente propiedades de configuración del clúster y del agente que no requieren el reinicio del agente.

Propiedad	Description (Descripción)	Predeterminado
<code>advertise d.listeners</code>	Oyentes que se publican para que los clientes los utilicen, si son distintos de la propiedad de configuración <code>listeners</code>. En entornos de IaaS, este valor puede necesitar ser distinto de la interfaz a la que se vincula el agente. Si no se establece, se utilizará el valor de oyentes. A diferencia de oyentes, no es válido anunciar la meta-dirección 0.0.0.0. Además, a diferencia de <code>listeners</code>, en esta propiedad pueden existir puertos	<code>null</code>

Propiedad	Description (Descripción)	Predeterminado
	duplicados, de modo que un oyente se pueda configurar para anunciar la dirección de otro oyente. Esto puede resultar útil en algunos casos en los que se utilizan equilibradores de carga externos. Esta propiedad se establece por agente.	

Propiedad	Description (Descripción)	Predeterminado
<code>compression.type</code>	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 <code>uncompressed</code> . Este valor equivale a la ausencia de compresión. Si establece el valor en <code>producer</code> , se retendrá el códec de compresión original que configuró el productor.	Valor predeterminado de Apache Kafka

Propiedad	Description (Descripción)	Predeterminado
log.cleaner.delete.retention.ms	La cantidad de tiempo durante la cual se conservan los marcadores de eliminación para los temas con registros compactados. Esta configuración también establece un límite de tiempo dentro del cual un consumidor debe completar una lectura si comienza desde el desplazamiento 0, para garantizar que obtenga una instantánea válida de la etapa final. De lo contrario, los marcadores de eliminación se podrían recopilar antes de que el	86400000 (24 * 60 * 60 * 1000 ms, es decir, 1 día), valor predeterminado de Apache Kafka

Propiedad	Description (Descripción)	Predeterminado
	consumidor complete su análisis.	
log.cleaner.min.compaction.lag.ms	El tiempo mínimo durante el cual un mensaje permanece sin compactar en el registro. Esta configuración solo se aplica a los registros que se compactan.	0, valor predeterminado de Apache Kafka

Propiedad	Description (Descripción)	Predeterminado
log.cleaner.max.compaction.lag.ms	El tiempo máximo durante el cual un mensaje permanece no apto para la compactación en el registro. Esta configuración solo se aplica a los registros que se compactan. Esta configuración estaría limitada en el rango de [7 días, Long.Max].	9223372036854775807, valor predeterminado de Apache Kafka

Propiedad	Description (Descripción)	Predeterminado
log.clean up.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 clústeres con almacenamiento por niveles habilitado, la política válida es únicamente delete.	Valor predeterminado de Apache Kafka

Propiedad	Description (Descripción)	Predeterminado
<code>log.message.timestamp.after.max.ms</code>	La diferencia de marca de tiempo permitida entre la marca de tiempo del mensaje y la marca de tiempo del agente. La marca de tiempo del mensaje puede ser posterior o igual a la marca de tiempo del agente, con la diferencia máxima permitida determinada por el valor establecido en esta configuración. Si <code>log.message.timestamp.type=CreateTime</code> , el mensaje se rechazará si la diferencia	86400000 (24 * 60 * 60 * 1000 ms, es decir, 1 día)

Propiedad	Description (Descripción)	Predeterminado
	a entre las marcas de tiempo supera este umbral especificado. Esta configuración se ignora si <code>log.message.timestamp.type=LogAppendTime</code> .	

Propiedad	Description (Descripción)	Predeterminado
<code>log.message.timestamp.before.max.ms</code>	La diferencia de marca de tiempo permitida entre la marca de tiempo del agente y la marca de tiempo del mensaje. La marca de tiempo del mensaje puede ser anterior o igual a la marca de tiempo del agente, con la diferencia máxima permitida determinada por el valor establecido en esta configuración. Si <code>log.message.timestamp.type=CreateTime</code> , el mensaje se rechazará si la diferenc	86400000 (24 * 60 * 60 * 1000 ms, es decir, 1 día)

Propiedad	Description (Descripción)	Predeterminado
	a entre las marcas de tiempo supera este umbral especificado. Esta configuración se ignora si <code>log.message.timestamp.type=LogAppendTime</code> .	
<code>log.message.timestamp.type</code>	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 <code>CreateTime</code> y <code>LogAppendTime</code> .	Valor predeterminado de Apache Kafka
<code>log.retention.bytes</code>	Tamaño máximo del registro antes de eliminarlo.	Valor predeterminado de Apache Kafka

Propiedad	Description (Descripción)	Predeterminado
log.retention.ms	Cantidad de milisegundos durante los cuales se conserva un archivo de registro antes de eliminarlo.	Valor predeterminado de Apache Kafka
max.connection.creation.rate	La tasa máxima de creación de conexiones permitida en el agente en un momento dado.	Valor predeterminado de Apache Kafka
max.connections	Número máximo de conexiones permitidas en el agente en un momento dado. Este límite se aplica además de cualquier límite por dirección IP configurado mediante <code>max.connections.per.ip</code> .	Valor predeterminado de Apache Kafka

Propiedad	Description (Descripción)	Predeterminado
max.connections.per.ip	El número máximo de conexiones permitidas desde cada dirección IP. Este valor se puede establecer en 0 si existen anulaciones configuradas mediante la propiedad max.connections.per.ip.overrides. Las nuevas conexiones procedentes de la dirección IP se descartan cuando se alcanza el límite.	Valor predeterminado de Apache Kafka

Propiedad	Description (Descripción)	Predeterminado
max.connections.per.ip.overrides	Lista separada por comas de anulaciones por dirección IP o nombre de host que sustituyen el número máximo de conexiones predeterminado. Un valor de ejemplo es hostName: 100,127.0.0.1:200	Valor predeterminado de Apache Kafka

Propiedad	Description (Descripción)	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	Valor predeterminado de Apache Kafka

Propiedad	Description (Descripción)	Predeterminado
	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 mediante la configuración <code>max.message.bytes</code> del tema.	

Propiedad	Description (Descripción)	Predeterminado
producer. id.expiration.ms	El tiempo, en milisegundos, que un líder de partición de un tema esperará antes de que caduquen los ID de productor . Los ID de productor no caducarán mientras una transacción asociada a estos siga en curso. Tenga en cuenta que los ID de productor pueden caducar antes si la última escritura realizada con ese ID de productor se elimina debido a la configuración de retención del tema. Establecer este valor igual	Valor predeterminado de Apache Kafka

Propiedad	Description (Descripción)	Predeterminado
	o superior a <code>delivery.timeout.ms</code> puede ayudar a evitar la caducidad durante los reintentos y a proteger frente a la duplicación de mensajes, pero el valor predeterminado debería ser razonable para la mayoría de los casos de uso.	

Topic-level configuraciones 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 se proporciona ninguna configuración de tema, Amazon MSK utiliza el valor predeterminado del agente. Al igual que con las configuraciones de agente, Amazon MSK protege algunas propiedades de configuración de temas frente a cambios. Los ejemplos incluyen el factor de replicación, `min.insync.replicas` y `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 de 3 de forma predeterminada. Para obtener más información sobre las propiedades de configuración a nivel de tema y ejemplos sobre cómo configurarlas, consulte las [Topic-Level configuraciones](#) en la documentación de Apache Kafka.

Propiedad	Description (Descripción)
<code>cleanup.policy</code>	Esta configuración designa la política de retención que se utilizará en los segmentos de registro. La política de “eliminar” (que es la predeterminada) descartará los segmentos antiguos cuando se alcance su límite de tiempo o de tamaño de retención. La política de “compactar” habilitará la compactación de registros, que conserva el valor más reciente de cada clave. También es posible especificar 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 agentes Express se activa una vez que los datos de una partición alcanzan los 256 MB.
<code>compression.type</code>	Especifique el tipo de compresión final para un tema determinado. Esta configuración acepta los códecs de compresión estándar (gzip, snappy, lz4 y zstd). Además, acepta <code>uncompressed</code> , que es equivalente a no usar compresión, y <code>producer</code> , que indica que se debe retener el códec de compresión original establecido por el productor.
<code>delete.retention.ms</code>	La cantidad de tiempo durante la cual se conservan los marcadores de eliminación para los temas con registros compactados. Esta configuración también establece un límite de tiempo dentro del cual un consumidor debe completar una lectura si comienza desde el desplazamiento 0, para garantizar que obtenga una instantánea válida de la etapa final. De lo

Propiedad	Description (Descripción)
	contrario, los marcadores de eliminación se podrían recopilar antes de que el consumidor complete su análisis. El valor predeterminado de esta configuración es 86400000 (24 × 60 × 60 × 1000 ms, es decir, 1 día). Valor predeterminado de Apache Kafka.
max.message.bytes	El tamaño máximo permitido de un lote de registros por Kafka (después de la compresión, si la compresión está habilitada). Si este valor se incrementa y existen consumidores con una versión anterior a 0.10.2, también se debe aumentar el tamaño de obtención de los consumidores para que puedan recuperar lotes de registros 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 anteriores, los registros sin comprimir no se agrupan en lotes y este límite se aplica únicamente a un solo registro en este caso. Este valor se puede configurar por tema mediante la <code>max.message.bytes</code> config del tema.

Propiedad	Description (Descripción)
<code>message.timestamp.after.max.ms</code>	Esta configuración establece la diferencia de tiempo permitida entre la marca de tiempo del mensaje y la marca de tiempo del agente. La marca de tiempo del mensaje puede ser posterior o igual a la marca de tiempo del agente, con la diferencia máxima permitida determinada por el valor establecido en esta configuración. Si <code>message.timestamp.type=CreateTime</code> , el mensaje se rechazará si la diferencia entre las marcas de tiempo supera este umbral especificado. Esta configuración se ignora si <code>message.timestamp.type=LogAppendTime</code> .
<code>message.timestamp.before.max.ms</code>	Esta configuración establece la diferencia de marca de tiempo permitida entre la marca de tiempo del agente y la marca de tiempo del mensaje. La marca de tiempo del mensaje puede ser anterior o igual a la marca de tiempo del agente, con la diferencia máxima permitida determinada por el valor establecido en esta configuración. Si <code>message.timestamp.type=CreateTime</code> , el mensaje se rechazará si la diferencia entre las marcas de tiempo supera este umbral especificado. Esta configuración se ignora si <code>message.timestamp.type=LogAppendTime</code> .
<code>message.timestamp.type</code>	Defina si la marca de tiempo del mensaje corresponde al momento de creación del mensaje o al momento de incorporación al registro. El valor debe ser <code>CreateTime</code> o <code>LogAppendTime</code> .

Propiedad	Description (Descripción)
min.compaction.lag.ms	El tiempo mínimo durante el cual un mensaje permanece sin compactar en el registro. Esta configuración solo se aplica a los registros que se compactan. El valor predeterminado de esta configuración es 0, valor predeterminado de Apache Kafka.
max.compaction.lag.ms	El tiempo máximo durante el cual un mensaje permanece no apto para la compactación en el registro. Esta configuración solo se aplica a los registros que se compactan. Esta configuración estaría limitada en el intervalo de [7 días,]. Long.Max El valor predeterminado de esta configuración es 9223372036854775807, el valor predeterminado de Apache Kafka.

Propiedad	Description (Descripción)
<code>retention.bytes</code>	Esta configuración controla el tamaño máximo que puede alcanzar una partición (que consta de segmentos de registro) antes de que se descarten los segmentos de registro antiguos para liberar espacio, cuando se utiliza la política de retención “eliminar”. De forma predeterminada, no existe un límite de tamaño, solo un límite de tiempo. Dado que este límite se aplica por partición, multiplíquelo por el número de particiones para calcular la retención del tema en bytes. Además, <code>retention.bytes</code> funciona de forma independiente de las configuraciones <code>segment.ms</code> y <code>segment.bytes</code>. Asimismo, esta configuración desencadena la rotación de un nuevo segmento si <code>retention.bytes</code> se configura con el valor 0.
<code>retention.ms</code>	Esta configuración controla el tiempo máximo durante el cual se retendrá un registro antes de descartar los segmentos antiguos para liberar espacio, cuando se utiliza la política de retención “eliminar”. Esto representa 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, la configuración <code>retention.ms</code> funciona de forma independiente de las configuraciones <code>segment.ms</code> y <code>segment.bytes</code>. Asimismo, esta configuración desencadena la rotación de un nuevo segmento si se cumple la condición <code>retention.ms</code>.

Configuraciones de solo lectura para agentes Express

Amazon MSK establece los valores de estas configuraciones y los protege frente a cambios que podrían afectar la disponibilidad del clúster. Estos valores pueden variar en función de la versión de Apache Kafka que se ejecute en el clúster, por lo que recuerde comprobar los valores correspondientes al clúster específico.

La siguiente tabla enumera las configuraciones de solo lectura para los agentes Express.

Propiedad	Description (Descripción)	Valor de agente Express
broker.id	El identificador del agente para este servidor.	1,2,3...
broker.rack	Rack del agente. Se utiliza en la asignación de réplicas con reconocimiento de rack para tolerancia a errores. Ejemplos: "RACK1", "us-east-1d"	ID de la zona de disponibilidad (AZ) o ID de la subred
default.replication.factor	Factores de replicación predeterminados para todos los temas	3
fetch.max.bytes	El número máximo de bytes que se devolverán para una solicitud de obtención.	Valor predeterminado de Apache Kafka
group.max.size	El número máximo de consumidores que puede admitir un único grupo de consumidores.	Valor predeterminado de Apache Kafka
inter.broker.listener.name	Nombre del oyente utilizado para la comunicación entre agentes.	REPLICATION_SECURE o REPLICATION

Propiedad	Description (Descripción)	Valor de agente Express
inter.broker.protocol.version	Especifica qué versión del protocolo de comunicación entre agentes se utiliza.	Valor predeterminado de Apache Kafka
oyentes	Lista de oyentes: Comma-separated lista de URIs que escucharemos y los nombres de los oyentes. Puede configurar la propiedad <code>advertised.listeners property</code> , pero no la propiedad <code>listeners</code> .	MSK-generated
log.message.format.version	Especifique la versión del formato de mensajes que el agente utilizará para anexar mensajes a los registros.	Valor predeterminado de Apache Kafka

Propiedad	Description (Descripción)	Valor de agente Express
<code>min.insync.replicas</code>	Cuando un productor establece <code>acks</code> en <code>all</code> (o <code>-1</code>), el valor en <code>min.insync.replicas</code> especifica el número mínimo de réplicas que deben confirmar una escritura para que esta se considere exitosa. Si no se puede cumplir este mínimo, el productor genera una excepción (ya sea <code>NotEnoughReplicas</code> o <code>NotEnoughReplicasAfterAppend</code>). Puede usar el valor de <code>acks</code> desde el productor para aplicar garantías de durabilidad más estrictas. Para ello, establezca <code>acks</code> en "todos". Esto garantiza que el productor emita una excepción si la mayoría de las réplicas no reciben una escritura.	2
<code>num.io.threads</code>	Número de subprocesos que el servidor utiliza para generar solicitudes, entre los que se puede incluir el disco. I/O (m7g.large, 8), (m7g.xlarge, 8), (m7g.2xlarge, 16), (m7g.4xlarge, 32), (m7g.8xlarge, 64), (m7g.12xlarge, 96), (m7g.16xlarge, 128)	Según el tipo de instancia. = $\text{Math.max}(8, 2 * \text{vCPU})$

Propiedad	Description (Descripción)	Valor de agente Express
num.network.threads	Número de subprocesos que el servidor usa para recibir solicitudes desde 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, vCPU)
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 <code>message.max.bytes</code> (configuración del agente) o <code>max.message.bytes</code> (configuración del tema) especifican el tamaño máximo del lote de registros que acepta el agente.	Valor predeterminado de Apache Kafka

Propiedad	Description (Descripción)	Valor de agente Express
<code>request.timeout.ms</code>	Esta configuración controla el tiempo máximo que el cliente esperará la respuesta de una solicitud. Si la respuesta no se recibe antes de que expire el tiempo de espera, el cliente reenviará la solicitud si es necesario o la marcará como fallida si se agotan los reintentos.	Valor predeterminado de Apache Kafka
<code>transaction.state.log.min.isr</code>	Se sobrescribe la configuración <code>min.insync.replicas</code> para el tema de transacciones.	2
<code>transaction.state.log.replication.factor</code>	El factor de replicación del tema de transacción.	Valor predeterminado de Apache Kafka
<code>unclean.leader.election.enable</code>	Permite que réplicas que no estén en el conjunto ISR actúen como líder como último recurso, aunque esto puede provocar pérdida de datos.	FALSO

Operaciones de configuración de agentes

Las configuraciones de los agentes de Apache Kafka pueden ser estáticas o dinámicas. Las configuraciones estáticas requieren reiniciar el agente para que los cambios se apliquen. Las configuraciones dinámicas no requieren reiniciar el agente 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](#)
- [Eliminación de una configuración](#)
- [Obtención de metadatos de configuración](#)
- [Obtención de detalles sobre una revisión de la configuración](#)
- [Enumeración de las configuraciones de la 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 al 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] [0-9 A-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 al 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
  }
}
```

Eliminación de una 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"
}
```



```
}
```

Obtención de 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"
}
```

Obtención de detalles sobre una 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
```

Enumeración de las configuraciones de la 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 comando siguiente.

```
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

Reequilibrio inteligente para clústeres

Amazon MSK proporciona un reequilibrio inteligente para todos los nuevos clústeres aprovisionados de MSK con agentes Express. Esta función administra automáticamente las operaciones de distribución de particiones y escalado de clústeres, lo que elimina la necesidad de utilizar herramientas de terceros. El reequilibrio inteligente reequilibra automáticamente las particiones al escalar los clústeres hacia arriba o hacia abajo. También supervisa de forma continua el estado del clúster para detectar desequilibrios o sobrecargas de recursos y redistribuye la carga de trabajo.

El reequilibrio inteligente proporciona operaciones de escalado rápidas que se completan en 30 minutos y no afecta a la disponibilidad del clúster durante el escalado. Está activado de forma predeterminada para todos los nuevos clústeres aprovisionados basados en MSK Express y funciona con el límite máximo de particiones recomendado de 20 000 particiones por agente. Además, esta función está disponible sin coste adicional y no requiere ninguna configuración.

A partir del 20 de noviembre de 2025, el reequilibrio inteligente estará disponible en todas AWS las regiones en las que se admita a los agentes de Amazon MSK Express.

Temas

- [Cómo funciona el reequilibrio inteligente](#)
- [Supervisión de las métricas de reequilibrio inteligente](#)
- [Consideraciones sobre el uso del reequilibrio inteligente](#)
- [Ampliar y reducir los clústeres de Amazon MSK con una sola operación](#)
- [Reequilibrio de estado estacionario para clústeres de Amazon MSK](#)

Cómo funciona el reequilibrio inteligente

El reequilibrio inteligente está activado de forma predeterminada en todos los nuevos clústeres aprovisionados de MSK con agentes Express. Incluye soporte para las siguientes situaciones:

- **Ampliación y reducción:** le permite añadir o eliminar agentes de sus clústeres basados en MSK Express con un solo clic. Una vez que especifique qué agentes desea añadir o eliminar, el reequilibrio inteligente redistribuye automáticamente las particiones en la nueva configuración del clúster según las mejores prácticas internas. AWS
- **Reequilibrio en estado estable:** en estado estable, esta función supervisa el estado del clúster de forma continua y reequilibra automáticamente las particiones cuando:
 - La utilización de los recursos se vuelve sesgada entre los corredores.
 - Los corredores se aprovisionan en exceso o se infrautilizan.
 - Se añaden nuevos corredores o se eliminan los corredores existentes.

Note

Si el reequilibrio inteligente está activado, no podrás utilizar herramientas de terceros, como Cruise Control, para reequilibrar las particiones. Primero debes pausar el reequilibrio inteligente para usar la API de reasignación de particiones proporcionada por estas herramientas de terceros.

Puede utilizar esta función en la consola de Amazon MSK. También puede utilizar esta función mediante Amazon MSK APIs o AWS SDK, y AWS CloudFormation. AWS CLI Para obtener más información, consulte [Escalar los clústeres de Amazon MSK](#) y [Reequilibrio en estado estacionario](#).

Supervisión de las métricas de reequilibrio inteligente

Puede supervisar el estado de las operaciones de reequilibrio inteligente en curso e históricas mediante las siguientes CloudWatch métricas de Amazon:

- **RebalanceInProgress:** Esta métrica se publica cada minuto con un valor de 1 cuando el reequilibrio está en curso y de 0 en caso contrario.
- **UnderProvisioned:** Indica que un clúster está actualmente insuficientemente aprovisionado y que no se puede realizar ningún reequilibrio de particiones. Tienes que añadir más agentes o ampliar el tipo de instancia del clúster.

Para obtener información sobre la supervisión de un clúster aprovisionado por MSK, consulte y. [Supervisión de un clúster de Amazon MSK aprovisionado](#) [Vea las métricas de Amazon MSK mediante CloudWatch](#)

Consideraciones sobre el uso del reequilibrio inteligente

- El soporte para el reequilibrio inteligente solo está disponible para los nuevos clústeres aprovisionados de MSK con corredores Express.
- Para la reasignación automática de particiones, está disponible un máximo de 20 000 particiones por agente.
- No puede utilizar herramientas de reasignación de particiones APIs ni de reequilibrio de terceros cuando el reequilibrio inteligente está activado. Para utilizar estas herramientas APIs o las de terceros, primero debe pausar el reequilibrio inteligente del clúster basado en MSK Express.

Ampliar y reducir los clústeres de Amazon MSK con una sola operación

Con el reequilibrio inteligente, puede escalar sus clústeres hacia arriba o hacia abajo editando el recuento de agentes en sus clústeres con una sola acción. Puede hacerlo en la consola de Amazon MSK o mediante Amazon MSK APIs o AWS SDK, y. AWS CLI AWS CloudFormation Al cambiar el recuento de agentes, Amazon MSK hace lo siguiente:

- Distribuye automáticamente las particiones a los nuevos corredores.
- Mueve las particiones de los corredores que se están eliminando.

A medida que amplía y reduce los clústeres, la disponibilidad de los clústeres para que los clientes produzcan y consuman datos no se ve afectada.

Scaling clusters using Consola de administración de AWS

1. ¿Abrir la consola Amazon MSK en <https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/>.
2. En la página Clústeres, elija un clúster basado en Express recién creado. Para obtener información sobre cómo crear un clúster aprovisionado basado en Express, consulte. [Paso 1: Creación de un clúster de MSK aprovisionado](#)
3. En la lista desplegable Acciones, seleccione Editar número de corredores.
4. En la página Editar el número de corredores por zona, realice una de las siguientes acciones:

- Para añadir más agentes a su clúster, seleccione Añadir agentes a cada zona de disponibilidad y, a continuación, introduzca el número de agentes que desee añadir.
- Para eliminar corredores de su clúster, elija Eliminar un corredor de cada zona de disponibilidad.

5. Seleccione Save changes (Guardar cambios).

Scaling clusters using AWS CLI

Puede ampliar o reducir la escala de sus clústeres editando su número de agentes. Para ello AWS CLI, utilice el [update-broker-count](#) comando, como se muestra en el siguiente ejemplo. En este comando, especifique en el target-broker-count parámetro el número de corredores que desea incluir en su clúster.

```
aws msk update-broker-count --cluster-arn arn:aws:kafka:us-east-1:123456789012:cluster/myCluster/abcd1234-5678-90ef-ghij-klmnopqrstuv-1 --current-version ABCDEF1GHIJK0L --target-broker-count 6
```

Scaling clusters using AWS SDK

Puede escalar sus clústeres hacia arriba o hacia abajo editando mediante programación el recuento de corredores. Para ello, utilice el AWS SDK, utilice la [UpdateBrokerCount](#) API, como se muestra en el siguiente ejemplo. Para el TargetNumberOfBrokerNodes parámetro, especifique el número de corredores que desea tener en su clúster.

```
update_broker_count_response = client.update_broker_count(
    ClusterArn='arn:aws:kafka:us-east-1:123456789012:cluster/myCluster/abcd1234-5678-90ef-ghij-klmnopqrstuv-1',
    CurrentVersion='ABCDEF1GHIJK0L',
    TargetNumberOfBrokerNodes=6
)
```

Reequilibrio de estado estacionario para clústeres de Amazon MSK

El reequilibrio en estado estable forma parte de la función de reequilibrio inteligente, que está activada de forma predeterminada en todos los nuevos clústeres aprovisionados por MSK con los intermediarios Express. A medida que amplía o reduce los clústeres, Amazon MSK gestiona automáticamente la administración de las particiones distribuyéndolas a nuevos agentes y

trasladándolas de los agentes para su eliminación. Para garantizar una distribución óptima de la carga de trabajo entre los corredores, el reequilibrio inteligente utiliza las mejores prácticas de Amazon MSK para determinar los umbrales para iniciar automáticamente el reequilibrio para sus corredores.

Puede pausar y reanudar el reequilibrio en estado estable cuando sea necesario. El reequilibrio en estado estacionario monitorea continuamente el clúster y hace lo siguiente:

- Realiza un seguimiento del uso de los recursos del bróker (CPU, red, almacenamiento).
- Ajusta la ubicación de las particiones automáticamente sin que ello afecte a la disponibilidad de los datos.
- Completa las operaciones de reequilibrio hasta 180 veces más rápido para los corredores Express en comparación con los corredores estándar.
- Mantiene el rendimiento del clúster.

Pause and resume steady state rebalancing in Consola de administración de AWS

1. ¿Abrir la consola Amazon MSK en <https://console.aws.amazon.com/msk/casa?region=us-east-1#/home/>.
2. En la página Clústeres, elija un clúster basado en Express. Para obtener información sobre cómo crear un clúster aprovisionado basado en Express, consulte. [Paso 1: Creación de un clúster de MSK aprovisionado](#)
3. En la página de detalles del clúster, compruebe que el estado de reequilibrio inteligente sea Activo. Si el reequilibrio inteligente no está disponible o el estado es En pausa, cree un nuevo clúster basado en Express.
4. En la lista desplegable Acciones, seleccione Editar el reequilibrio inteligente.
5. En la página Editar el reequilibrio inteligente, haga lo siguiente:
 - a. Seleccione En pausa.
 - b. Seleccione Save changes (Guardar cambios).

Pause and resume steady state rebalancing using AWS CLI

Para establecer el estado de reequilibrio de un clúster en el que se **ACTIVE** utilice el AWS CLI, utilice el comando [update-rebalancing](#), como se muestra en el siguiente ejemplo. En este comando, especifique el estado con el parámetro. `rebalancing`


```
aws msk update-rebalancing --cluster-arn arn:aws:kafka:us-
east-1:123456789012:cluster/myCluster/abcd1234-5678-90ef-ghij-klmnopqrstuv-1
--current-version ABCDEF1GHIJK0L --rebalancing "{\"Rebalancing\":{\"Status\":
\\\"ACTIVE\\\"}}"
```

Pause and resume steady state rebalancing using AWS SDK

También puede establecer el estado de reequilibrio de un clúster mediante la [UpdateRebalancingRequest](#) API para modificar mediante programación el recuento de agentes. Los siguientes ejemplos muestran cómo establecer el estado de reequilibrio en **ACTIVE** y **PAUSED**.

```
final UpdateRebalancingRequest updateRebalancingRequest = new
UpdateRebalancingRequest()
    .withClusterArn(arn:aws:kafka:us-
east-1:123456789012:cluster/myCluster/abcd1234-5678-90ef-ghij-klmnopqrstuv-1)
    .withCurrentVersion(ABCDEF1GHIJK0L)
    .withRebalancing(new Rebalancing().withStatus("ACTIVE"));
```

```
final UpdateRebalancingRequest updateRebalancingRequest = new
UpdateRebalancingRequest()
    .withClusterArn(arn:aws:kafka:us-
east-1:123456789012:cluster/myCluster/abcd1234-5678-90ef-ghij-klmnopqrstuv-1)
    .withCurrentVersion(ABCDEF1GHIJK0L)
    .withRebalancing(new Rebalancing().withStatus("PAUSED"));
```

Aplicación de parches en clústeres de MSK aprovisionados

De forma periódica, Amazon MSK actualiza el software en los agentes del clúster. El mantenimiento incluye actualizaciones planificadas o reparaciones no planificadas. El mantenimiento planificado incluye actualizaciones del sistema operativo, actualizaciones de seguridad y otras actualizaciones de software necesarias para mantener el estado, la seguridad y el rendimiento del clúster. Realizamos mantenimiento no planificado para resolver degradaciones repentinas de la infraestructura. Realizamos mantenimiento tanto en agentes Standard como en agentes Express, aunque las experiencias son diferentes.

Aplicación de parches en agentes Standard

Las actualizaciones de los agentes Standard no afectan las operaciones de escritura y lectura de las aplicaciones si sigue las [prácticas recomendadas](#).

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. Al ver las operaciones del clúster en Consola de administración de AWS o a través de `DescribeClusterOperation` y `ListClusterOperations` APIs, estas operaciones de mantenimiento aparecen con un tipo de operación de `SECURITY_PATCHING`. 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 [Prácticas recomendadas para clientes de Apache Kafka](#) para garantizar un funcionamiento fluido del clúster en todo momento, incluso durante los procesos de 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 esperado, ya que las particiones del agente que se apagó dejan de replicar 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.

Aplicación de parches para agentes Express

No existen periodos de mantenimiento para los agentes Express. Amazon MSK actualiza automáticamente el clúster de forma continua y distribuida en el tiempo, lo que significa que puede

esperar reinicios ocasionales y puntuales de agentes a lo largo del mes. Esto garantiza que no necesite planificar ni realizar ajustes en torno a periodos de mantenimiento únicos de los clústeres. Como siempre, el tráfico permanecerá ininterrumpido durante el reinicio de un agente, ya que el liderazgo se trasladará a otros agentes que continuarán atendiendo las solicitudes. Al ver las operaciones del Consola de administración de AWS clúster en `DescribeClusterOperation` y a través de ella `ListClusterOperations` APIs, estas operaciones de mantenimiento aparecen con un tipo de operación `deBROKER_UPDATE`.

Los agentes Express se entregan configurados con ajustes y barreras de protección basados en prácticas recomendadas que hacen que el clúster sea resiliente frente a cambios de carga que se puedan producir durante el mantenimiento. Amazon MSK establece cuotas de rendimiento en los agentes Express para mitigar el impacto de una sobrecarga del clúster, que podría provocar problemas durante los reinicios de agentes. Estas mejoras eliminan la necesidad de notificaciones anticipadas, planificación previa y periodos de mantenimiento cuando se utilizan agentes Express.

Los agentes Express siempre replican los datos en tres copias, por lo que los clientes realizan la conmutación por error de forma automática durante los reinicios. No es necesario preocuparse por que los temas queden inaccesibles, incluso cuando el factor de replicación está configurado en 1 o 2. Además, la sincronización posterior de un agente Express que se reinicia es más rápida que en el caso de los agentes Standard. La mayor velocidad de aplicación de parches en los agentes Express implica una interrupción mínima en la planificación de cualquier actividad del plano de control que tenga programada para el clúster.

Como ocurre con todas las aplicaciones de Apache Kafka, se mantiene un contrato compartido cliente-servidor para los clientes que se conectan a agentes Express. Resulta fundamental configurar los clientes para que puedan gestionar correctamente la conmutación por error del liderazgo entre los agentes. Siga las [Prácticas recomendadas para clientes de Apache Kafka](#) para garantizar un funcionamiento fluido del clúster en todo momento, incluso durante la aplicación de parches. Tras el reinicio de un agente, es normal que los clientes experimenten [errores transitorios de desconexión](#). Esto no afecta a las operaciones de producción ni de consumo, ya que los agentes seguidores asumen el liderazgo de las particiones. Sus clientes de Apache Kafka realizarán automáticamente la conmutación por error y comenzarán 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 (normalmente es alta en milisegundos en 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.

Resolución 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 deben alarmarse ante las excepciones transitorias, `network/not_leader` ya que son normales, no impactan y se espera que formen parte del protocolo Kafka.
- Los clientes no deben activar la alarma, `UnderReplicatedPartitions` ya que es normal, no tienen impacto y se espera que se produzcan cuando se trata de un intermediario fuera de línea.

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 <https://aws.amazon.com/compliance/shared-responsibility-model/> se refiere a estos conceptos 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. Third-party los auditores comprueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los programas de [AWS cumplimiento, los programas](#) 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 de las API de Amazon MSK](#)
- [Autenticación y autorización para las API de Apache Kafka](#)
- [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 la 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, consulte las [Preguntas frecuentes sobre privacidad de datos](#) y los . Para obtener más información sobre la protección de datos en Europa, consulte el [Centro del Reglamento General de Protección de Datos \(RGPD\)](#).

Para fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure 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. Exigimos 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 la información confidencial almacenada 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, consulte [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 dispositivo Servicios de AWS mediante la consola, la API o los AWS SDK. AWS CLI Cualquier dato que introduzca en etiquetas o campos de formato libre utilizados para los nombres se pueden 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](#)
- [Uso de las API de Amazon MSK con puntos de conexión 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 con agentes Express permanecen en el estado ACTIVE cuando Amazon MSK inicia la operación de actualización de certificados. En los clústeres con agentes Standard, Amazon MSK establece el estado del clúster en MAINTENANCE cuando inicia la operación de actualización de certificados. MSK restablece el estado a ACTIVE cuando finaliza la actualización. Mientras un clúster se encuentra en la operación de actualización de certificados, puede continuar con la producción y el consumo de datos, pero no puede realizar ninguna operación de actualización en este.

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. Esta es la configuración predeterminada.
- 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 [Determinación de 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](#).

Uso de las API de Amazon MSK con puntos de conexión 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 las API de Amazon 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 PrivateLink](#) es una AWS tecnología que permite la comunicación privada entre AWS servicios mediante una interfaz de red elástica con direcciones IP privadas 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 las API Amazon MSK Provisioned y MSK Connect 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. FIPS-enabled Los puntos finales de VPC de interfaz están disponibles para las regiones de EE. UU. Para obtener más información, consulte [Creación de un punto de conexión de interfaz](#).

Al utilizar esta característica, los clientes de Apache Kafka pueden recuperar dinámicamente las cadenas de conexión necesarias para conectarse a los recursos de MSK aprovisionado o MSK Connect, sin atravesar Internet para obtener dichas cadenas de conexión.

Cuando cree un punto de conexión de VPC de interfaz, elija uno de los siguientes puntos de conexión de nombre de servicio:

Para MSK aprovisionado:

- Los siguientes puntos finales con nombres de servicio ya no se admiten para las nuevas conexiones:
 - com.amazonaws.region.kafka

- `com.amazonaws.region.kafka-fips ()` FIPS-enabled
- El servicio de punto final de Dualstack que admite tráfico IPv4 e IPv6 es:
 - `aws.api.region.kafka-api`
 - `aws.api.region.kafka-api-fips ()` FIPS-enabled

Para configurar los puntos finales de doble pila, debe seguir las pautas de los puntos finales de FIPS. Dual-stack

Donde “region” es el nombre de la región. Elija este nombre de servicio para trabajar con las API de MSK. Provisioned-compatible Para obtener más información, consulte [Operaciones](https://docs.aws.amazon.com/msk/1.0/apireference/) en. <https://docs.aws.amazon.com/msk/1.0/apireference/>

Para MSK Connect:

- `com.amazonaws.region.kafkaconnect`

Donde “region” es el nombre de la región. Elija este nombre de servicio para trabajar con las Connect-compatible API de MSK. Para obtener más información, consulte [Acciones](#) en la Referencia de la API de Amazon MSK Connect.

Para obtener más información, incluidas instrucciones paso a paso para crear un punto de conexión de VPC de tipo interfaz, consulte [Creación de un punto de conexión de interfaz](#) en la Guía de AWS PrivateLink .

Control del acceso a los puntos de conexión de VPC para las API de MSK aprovisionado o MSK Connect

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. Utilice la política de ejemplo adecuada para definir los permisos de acceso para el servicio MSK aprovisionado o MSK Connect, según corresponda.

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 más información, consulte [Control del acceso a los servicios con puntos de conexión de VPC](#) en la Guía de AWS PrivateLink .

MSK Provisioned — VPC policy example

Read-only acceder

Esta política de ejemplo se puede asociar a un punto de conexión de VPC. (Para obtener más información, consulte [Control de acceso a recursos de Amazon VPC](#)). Restringe las acciones únicamente a las operaciones de listado y descripción a través del punto de conexión de VPC al que está asociada.

```
{
  "Statement": [
    {
      "Sid": "MSKReadOnly",
      "Principal": "*",
      "Action": [
        "kafka:List*",
        "kafka:Describe*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Ejemplo de política de punto de conexión de VPC para MSK aprovisionado

Restricción del acceso a un clúster de MSK específico

Esta política de ejemplo se puede asociar a un punto de conexión de VPC. Restringe el acceso a un clúster específico de Kafka a través del punto de conexión de VPC al que está asociada.

```
{
  "Statement": [
    {
      "Sid": "AccessToSpecificCluster",
      "Principal": "*",
      "Action": "kafka:*",
      "Effect": "Allow",
      "Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/MyCluster"
    }
  ]
}
```

```
]
}
```

MSK Connect — VPC endpoint policy example

Enumeración de conectores y creación de un conector nuevo

El siguiente es un ejemplo de una política de punto de conexión para MSK Connect. Esta política permite que el rol especificado enumere conectores y cree 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/MyMSKConnectExecutionRole"
        ]
      }
    }
  ]
}
```

Ejemplo de política de punto de conexión de VPC para MSK Connect

Permite únicamente solicitudes provenientes de una dirección IP específica dentro de 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 de las API de Amazon MSK

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a AWS los recursos. 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](#)
- [Service-linked funciones 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. 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.

Tenga en cuenta que las acciones políticas para las API temáticas de MSK utilizan el `kafka-cluster` prefijo antes de la acción; consulte la [Semántica de las acciones y los recursos de la política de autorización de IAM](#)

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 puede 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. Como práctica recomendada, especifique un recurso utilizando el [Nombre de recurso de Amazon \(ARN\)](#). En el caso de las acciones que no admiten permisos por recurso, utilice 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 los ARN, consulte Nombres de [recursos de Amazon \(ARN\) y espacios de nombres de AWS servicio](#).

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 única instrucción, separe los ARN con comas.

```
"Resource": ["resource1", "resource2"]
```

Para ver una lista de los tipos de recursos de Amazon MSK y los ARN, consulte [Tipos de recurso definidos por Amazon Managed Streaming para Apache 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 puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` especifica cuándo se ejecutan las instrucciones en función de criterios definidos. Puede 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. 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.

El tamaño máximo admitido para una política de clúster es de 20 KB.

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 “Etiquetado de un clúster”](#).

Puede controlar el acceso al clúster únicamente con la ayuda de etiquetas. Para etiquetar temas y grupos de consumidores, debe agregar una declaración independiente en las políticas sin etiquetas.

Para ver un ejemplo de una política basada en identidad que limita 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 a un usuario describir el clúster, obtener sus agentes de arranque, enumerar sus nodos de agentes, actualizarlo y eliminarlo. Sin embargo, esta política concede permisos únicamente si la etiqueta del clúster `Owner` tiene el valor del `username` de ese usuario. La segunda declaración de la siguiente política permite el acceso a los temas del clúster. La primera declaración de esta política no autoriza ningún acceso a los temas.

JSON

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

```

    "Sid": "AccessClusterIfOwner",
    "Effect": "Allow",
    "Action": [
        "kafka:Describe*",
        "kafka:Get*",
        "kafka:List*",
        "kafka:Update*",
        "kafka:Delete*"
    ],
    "Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/Owner": "${aws:username}"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "kafka-cluster:*Topic*",
        "kafka-cluster:WriteData",
        "kafka-cluster:ReadData"
    ],
    "Resource": [
        "arn:aws:kafka:us-east-1:123456789012:topic/*"
    ]
}
]
}

```

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 [AssumeRole](#) o [GetFederationToken](#).

Amazon MSK admite el uso de credenciales temporales.

Service-linked roles

[Service-linked los roles](#) permiten a Amazon Web Services acceder a los recursos de otros servicios para completar una acción en su nombre. Service-linked los roles aparecen en su 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 “Service-linked funciones”](#).

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

Identity-based las políticas determinan si alguien puede crear, acceder o eliminar los recursos de Amazon MSK de su 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 para 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, consulte 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, consulte [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 CloudFormation. Para obtener más información, consulte [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 administradas 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.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UpdateCluster",
      "Effect": "Allow",
      "Action": [
        "kafka:Describe*",
        "kafka:Get*",
        "kafka:List*",
        "kafka:Update*"
      ],
      "Resource": "arn:aws:kafka:us-east-1:012345678012:cluster/
purchaseQueriesCluster/abcdefab-1234-abcd-5678-cdef0123ab01-2"
    }
  ]
}
```

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.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AccessClusterIfOwner",
      "Effect": "Allow",
      "Action": [
        "kafka:Describe*",
        "kafka:Get*",
        "kafka:List*",

```

```
    "kafka:Update*",
    "kafka:Delete"
  ],
  "Resource": "arn:aws:kafka:us-east-1:012345678012:cluster/*",
  "Condition": {
    "StringEquals": {
      "aws:ResourceTag/Owner": "${aws:username}"
    }
  }
}
```

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.

Service-linked funciones para Amazon MSK

Amazon MSK utiliza funciones vinculadas a [servicios AWS Identity and Access Management](#) (IAM). Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente a Amazon MSK. Service-linked Amazon MSK predefine los roles 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 sobre otros servicios que admiten funciones vinculadas a servicios, consulte [Amazon Web Services That Work with IAM](#) y busque los servicios que tengan la palabra **Sí** en la Service-Linked column Role. Elija una opción **Sí** con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Temas

- [Service-linked permisos de rol 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](#)

Service-linked permisos de rol 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.
- `*Ipv6Addresses`— asignar y desasignar direcciones IPv6 a las interfaces de red de la cuenta del cliente para habilitar la conectividad IPv6 para los clústeres de MSK.
- `ModifyNetworkInterfaceAttribute`— modifique los atributos de la interfaz de red en la cuenta del cliente para configurar los ajustes de IPv6 para la conectividad de los clústeres de MSK.

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.

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 [los permisos de Service-Linked rol](#) 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 Consola de administración de AWS, 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 Service-Linked rol](#) 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. 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 por AWS](#) en la Guía del usuario de IAM.

AWS política gestionada: AmazonMSKFullAccess

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 permisos de Amazon EC2 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.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "kafka:*",
      "ec2:DescribeSubnets",
      "ec2:DescribeVpcs",
      "ec2:DescribeSecurityGroups",
      "ec2:DescribeRouteTables",
```

```

    "ec2:DescribeVpcEndpoints",
    "ec2:DescribeVpcAttribute",
    "kms:DescribeKey",
    "kms:CreateGrant",
    "logs:CreateLogDelivery",
    "logs:GetLogDelivery",
    "logs:UpdateLogDelivery",
    "logs>DeleteLogDelivery",
    "logs:ListLogDeliveries",
    "logs:PutResourcePolicy",
    "logs:DescribeResourcePolicies",
    "logs:DescribeLogGroups",
    "S3:GetBucketPolicy",
    "firehose:TagDeliveryStream"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:CreateVpcEndpoint"
  ],
  "Resource": [
    "arn:*:ec2:*:*:vpc/*",
    "arn:*:ec2:*:*:subnet/*",
    "arn:*:ec2:*:*:security-group/*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:CreateVpcEndpoint"
  ],
  "Resource": [
    "arn:*:ec2:*:*:vpc-endpoint/*"
  ],
  "Condition": {
    "StringEquals": {
      "aws:RequestTag/AWSMSKManaged": "true"
    },
    "StringLike": {
      "aws:RequestTag/ClusterArn": "*"
    }
  }
}

```

```

    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:*:ec2:*:*:vpc-endpoint/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "CreateVpcEndpoint"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteVpcEndpoints"
      ],
      "Resource": "arn:*:ec2:*:*:vpc-endpoint/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/AWSMSKManaged": "true"
        },
        "StringLike": {
          "ec2:ResourceTag/ClusterArn": "*"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "kafka.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/kafka.amazonaws.com/AWSServiceRoleForKafka*",
      "Condition": {

```



```

    "StringLike": {
      "iam:AWSServiceName": "kafka.amazonaws.com"
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "iam:AttachRolePolicy",
      "iam:PutRolePolicy"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/kafka.amazonaws.com/
AWSServiceRoleForKafka*"
  },
  {
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "arn:aws:iam::*:role/aws-service-role/
delivery.logs.amazonaws.com/AWSServiceRoleForLogDelivery*",
    "Condition": {
      "StringLike": {
        "iam:AWSServiceName": "delivery.logs.amazonaws.com"
      }
    }
  }
]
}

```

AWS política gestionada: AmazonMSKReadOnlyAccess

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.

- Permisos de **Amazon EC2**: se utilizan para describir Amazon VPC, las subredes, los grupos de seguridad y las ENI asociadas a un clúster.
- Permisos de **AWS KMS**: se utiliza para describir la clave asociada al clúster.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "kafka:Describe*",
        "kafka:List*",
        "kafka:Get*",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "kms:DescribeKey"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

AWS 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 “Service-linked funciones”](#).

En la siguiente tabla se describen las actualizaciones de la política KafkaServiceRolePolicy gestionada desde que Amazon MSK comenzó a rastrear los cambios.

Cambio	Descripción	Fecha
Se agregó el soporte de conectividad IPv6 KafkaServiceRolePolicy : se actualizó a una política existente	Amazon MSK agregó permisos para habilitar la conectividad IPv6 KafkaServiceRolePolicy para los clústeres de MSK. Estos permisos permiten a Amazon MSK asignar y desasignar direcciones IPv6 a las interfaces de red y modificar los atributos de las interfaces de red en la cuenta del cliente.	17 de noviembre de 2025
KafkaServiceRolePolicy : actualización de una política actual	Amazon MSK agregó permisos para admitir la conectividad privada con varias VPC.	8 de marzo de 2023
Amazon MSK comenzó a hacer un seguimiento de los cambios	Amazon MSK ha empezado a realizar un seguimiento de los cambios de la política KafkaServiceRolePolicy gestionada.	8 de marzo de 2023

AWS política gestionada: AWSMSKReplicatorExecutionRole

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.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ClusterPermissions",
      "Effect": "Allow",
      "Action": [
        "kafka-cluster:Connect",
        "kafka-cluster:DescribeCluster",
        "kafka-cluster:AlterCluster",
        "kafka-cluster:DescribeTopic",
        "kafka-cluster:CreateTopic",
        "kafka-cluster:AlterTopic",
        "kafka-cluster:WriteData",
        "kafka-cluster:ReadData",
        "kafka-cluster:AlterGroup",
        "kafka-cluster:DescribeGroup",
        "kafka-cluster:DescribeTopicDynamicConfiguration",
        "kafka-cluster:AlterTopicDynamicConfiguration",
        "kafka-cluster:WriteDataIdempotently"
      ],
      "Resource": [
        "arn:aws:kafka:*:*:cluster/*"
      ]
    },
    {
      "Sid": "TopicPermissions",
      "Effect": "Allow",
      "Action": [
        "kafka-cluster:DescribeTopic",
        "kafka-cluster:CreateTopic",
        "kafka-cluster:AlterTopic",
        "kafka-cluster:WriteData",
        "kafka-cluster:ReadData",
        "kafka-cluster:DescribeTopicDynamicConfiguration",
        "kafka-cluster:AlterTopicDynamicConfiguration",

```

```
    "kafka-cluster:AlterCluster"
  ],
  "Resource": [
    "arn:aws:kafka:*:*:topic/*/*"
  ]
},
{
  "Sid": "GroupPermissions",
  "Effect": "Allow",
  "Action": [
    "kafka-cluster:AlterGroup",
    "kafka-cluster:DescribeGroup"
  ],
  "Resource": [
    "arn:aws:kafka:*:*:group/*/*"
  ]
}
]
```

Amazon MSK actualiza a AWS políticas administradas

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 AWSMSKReplicatorExecutionRole : actualización a una política existente	Amazon MSK agregó WriteDataIdempotently permisos a la AWSMSKReplicatorExecutionRole política para admitir la replicación de datos entre clústeres de MSK.	12 de marzo de 2024
AWSMSKReplicatorExecutionRole : política nueva	Amazon MSK agregó una AWSMSKReplicatorExecutionRole política para admitir Amazon MSK Replicator.	4 de diciembre de 2023

Cambio	Descripción	Fecha
AmazonMSKFullAccess: actualización de una política actual	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
AmazonMSKFullAccess: actualización de una política actual	Amazon MSK agregó nuevos permisos de Amazon EC2 para permitir la conexión a un clúster.	30 de noviembre de 2021
AmazonMSKFullAccess: actualización de una política actual	Amazon MSK agregó un nuevo permiso que le permite describir las tablas de enrutamiento de Amazon EC2.	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 Consola de administración de AWS le indica que no está autorizado a realizar una acción, debe ponerse en contacto con su administrador para obtener ayuda. El administrador 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 las API de Apache Kafka

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 las ACL de Apache Kafka 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 de las API de Amazon MSK”](#).

Temas

- [Control de acceso de IAM](#)
- [Autenticación TLS mutua de clientes para Amazon MSK](#)
- [Sign-in autenticación de credenciales con AWS Secrets Manager](#)
- [ACL de Apache Kafka](#)

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 mediante IAM para clientes que no son Java está disponible para clústeres de MSK con la versión 2.7.1 de Kafka o posterior.

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 las API de ACL de Apache Kafka para un clúster de MSK que utilice el control de acceso de IAM. Sin embargo, las ACL de Apache Kafka no tienen ningún efecto sobre la autorización de identidades de IAM. Debe utilizar políticas de IAM para controlar el acceso de las identidades de IAM.

Consideraciones importantes

Cuando utilice el control de acceso mediante IAM con el 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 las API de ACL de Apache Kafka para un clúster de MSK que utilice el control de acceso de IAM. Sin embargo, las ACL de Apache Kafka no tienen ningún efecto sobre la autorización de identidades de IAM. Debe utilizar 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 Consola de administración de AWS 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](#).

Use la Consola de administración de AWS 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 el 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 [CreateCluster](#) API 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:

- Non-Java configuración del cliente mediante el mecanismo SASL_OAUTHBEARER
- Configuración del cliente Java mediante el mecanismo o mecanismo SASL_OAUTHBEARER AWS_MSK_IAM

Uso del mecanismo SASL_OAUTHBEARER para configurar IAM

1. Edite el archivo de configuración `client.properties` con el siguiente ejemplo de cliente de Kafka en Python. 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 lenguaje de configuración que haya elegido y siga las instrucciones de la sección Introducción de la página principal de la biblioteca correspondiente.

- 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>
- Ir: <https://github.com/aws/aws-msk-iam-sasl-signer-go#getting-started>
- .NET: <https://github.com/aws/aws-msk-iam-sasl-signer-net#getting-started>
- JAVA: el soporte de SASL_OAUTHBEARER 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 JAR estable de [aws-msk-iam-auth](#) 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 su cliente está en una instancia de Amazon EC2, asocie la política de autorización al rol de IAM de esa instancia de Amazon EC2. 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](#).

A continuación se muestra 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 realice en una política de IAM se reflejan en las API de IAM y en la AWS CLI de manera inmediata. 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.

JSON

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

```

        "Action": [
            "kafka-cluster:Connect",
            "kafka-cluster:AlterCluster",
            "kafka-cluster:DescribeCluster"
        ],
        "Resource": [
            "arn:aws:kafka:us-east-1:111122223333:cluster/MyTestCluster/
abcd1234-0123-abcd-5678-1234abcd-1"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "kafka-cluster:*Topic*",
            "kafka-cluster:WriteData",
            "kafka-cluster:ReadData"
        ],
        "Resource": [
            "arn:aws:kafka:us-east-1:123456789012:topic/MyTestCluster/*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "kafka-cluster:AlterGroup",
            "kafka-cluster:DescribeGroup"
        ],
        "Resource": [
            "arn:aws:kafka:us-east-1:123456789012:group/MyTestCluster/*"
        ]
    }
]
}

```

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 `WriteDataIdempotently` permiso está obsoleto (). [KIP-679](#) Se utiliza `enable.idempotence = true` de forma predeterminada. Por lo tanto, para las versiones de Kafka 2.8.0 y posteriores, IAM no ofrece la misma funcionalidad que las ACL de Kafka. No es posible `WriteDataIdempotently` en un tema si se proporciona únicamente acceso de

WriteData a ese tema. Esto no afecta al caso en el que se proporciona WriteData 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 a la forma en que se implementan las ACL de Kafka. Además, la escritura idempotente en un tema también requiere acceso a `transactional-ids`.

Para solucionar esta limitación, se recomienda utilizar una política similar a la que se muestra a continuación.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kafka-cluster:Connect",
        "kafka-cluster:AlterCluster",
        "kafka-cluster:DescribeCluster",
        "kafka-cluster:WriteDataIdempotently"
      ],
      "Resource": [
        "arn:aws:kafka:us-east-1:123456789012:cluster/MyTestCluster/abcd1234-0123-abcd-5678-1234abcd-1"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kafka-cluster:*Topic*",
        "kafka-cluster:WriteData",
        "kafka-cluster:ReadData"
      ],
      "Resource": [
        "arn:aws:kafka:us-east-1:123456789012:topic/MyTestCluster/abcd1234-0123-abcd-5678-1234abcd-1/TestTopic",
        "arn:aws:kafka:us-east-1:123456789012:transactional-id/MyTestCluster/abcd1234-0123-abcd-5678-1234abcd-1/*"
      ]
    }
  ]
}
```

En este caso, `WriteData` permite escribir en `TestTopic` y, al mismo tiempo, `WriteDataIdempotently` permite escrituras idempotentes en el clúster. Esta política también agrega acceso a los recursos de `transactional-id` que serán necesarios.

Dado que `WriteDataIdempotently` es un permiso de clúster, no se puede utilizar en los temas. Si `WriteDataIdempotently` se restringe 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

Note

En el caso de los clústeres que ejecutan la versión 3.8 o posterior de Apache Kafka, el control de acceso de IAM admite la API para finalizar las transacciones `WriteTxnMarkers`. En el caso de los clústeres que ejecutan versiones de Kafka anteriores a la 3.8, el control de acceso de IAM no admite ni siquiera las acciones internas del clúster. `WriteTxnMarkers` En estas versiones anteriores, para finalizar las transacciones, utiliza la autenticación SCRAM o mTLS con las ACL adecuadas 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`.

Action	Description (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 DESCRIBE_	kafka-cluster:Connect	clúster	No

Action	Description (Descripción)	Acciones obligatorias	Recursos necesarios de	Aplicable a los clústeres sin servidor
	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 idempotente en un clúster, equivalente a la ACL de CLÚSTER IDEMPOTENT_WRITE de Apache Kafka.	kafka-cluster:Connect kafka-cluster:WriteData	clúster	Sí

Action	Description (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

Action	Description (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:DescribeTopic DynamicConfiguration	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:DescribeTopic DynamicConfiguration	tema	Sí

Action	Description (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

Action	Description (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 los ID de transacción en un clúster, equivalente a la ACL DESCRIBE TRANSACTIONAL_ID de Apache Kafka.	kafka-cluster:Connect	transactional-id	Sí

Action	Description (Descripción)	Acciones obligatorias	Recursos necesarios de	Aplicable a los clústeres sin servidor
kafka-cluster:AlterTransactionalId	Otorga permiso para modificar los ID de transacción en un clúster, equivalente 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 Consola de administración de AWS o mediante la [DescribeCluster](#) API o el comando [AWS CLI describe-cluster](#). A continuación, puede usar el ARN del clúster para crear los ARN de temas, grupos e ID de transacción. Para especificar un recurso en una política de autorización, se utiliza el ARN de ese recurso.

Recurso	Formato de ARN
Clúster	<code>arn:aws:kafka: <i>region</i> :<i>account-id</i> :<i>cluster-name</i> :<i>cluster-uuid</i></code>
Topic	<code>arn:aws:kafka: <i>region</i> :<i>account-id</i> :<i>cluster-name</i> :<i>cluster-uuid</i> :<i>topic-name</i></code>
Group	<code>arn:aws:kafka: <i>region</i> :<i>account-id</i> :<i>cluster-name</i> :<i>cluster-uuid</i> :<i>group-name</i></code>
ID de transacción	<code>arn:aws:kafka: <i>region</i> :<i>account-id</i> :<i>cluster-name</i> :<i>cluster-uuid</i> :<i>transactional-id</i></code>

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	<code>kafka-cluster:*</code>
Crear un tema	<code>kafka-cluster:Connect</code> <code>kafka-cluster:CreateTopic</code>
Producir datos	<code>kafka-cluster:Connect</code> <code>kafka-cluster:DescribeTopic</code> <code>kafka-cluster:WriteData</code>
Consumir datos	<code>kafka-cluster:Connect</code> <code>kafka-cluster:DescribeTopic</code> <code>kafka-cluster:DescribeGroup</code> <code>kafka-cluster:AlterGroup</code> <code>kafka-cluster:ReadData</code>
Producir datos de forma idempotente	<code>kafka-cluster:Connect</code>

Caso de uso	Acciones obligatorias
	<code>kafka-cluster:DescribeTopic</code> <code>kafka-cluster:WriteData</code> <code>kafka-cluster:WriteDataIdempotently</code>
Producir datos de forma transaccional	<code>kafka-cluster:Connect</code> <code>kafka-cluster:DescribeTopic</code> <code>kafka-cluster:WriteData</code> <code>kafka-cluster:DescribeTransactionalId</code> <code>kafka-cluster:AlterTransactionalId</code>
Describir la configuración de un clúster	<code>kafka-cluster:Connect</code> <code>kafka-cluster:DescribeClusterDynamicConfiguration</code>
Actualizar la configuración de un clúster	<code>kafka-cluster:Connect</code> <code>kafka-cluster:DescribeClusterDynamicConfiguration</code> <code>kafka-cluster:AlterClusterDynamicConfiguration</code>
Describir la configuración de un tema	<code>kafka-cluster:Connect</code> <code>kafka-cluster:DescribeTopicDynamicConfiguration</code>

Caso de uso	Acciones obligatorias
Actualizar la configuración de un tema	<code>kafka-cluster:Connect</code> <code>kafka-cluster:DescribeTopic</code> <code>DynamicConfiguration</code> <code>kafka-cluster:AlterTopicDynamicConfiguration</code>
Modificar un tema	<code>kafka-cluster:Connect</code> <code>kafka-cluster:DescribeTopic</code> <code>kafka-cluster:AlterTopic</code>

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 la Cuenta de AWS misma cuenta que su 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](#).

Amazon MSK no admite listas de revocación de certificados (CRL). Para controlar el acceso a los temas de su clúster o bloquear los certificados comprometidos, utilice las ACL y los grupos de AWS seguridad de Apache Kafka. Para obtener más información sobre cómo utilizar las ACL de Apache Kafka, consulte [the section called “ACL de Apache Kafka”](#).

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 por las PCA solo se autenticuen 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 el 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

En este proceso, se describe cómo configurar una instancia de Amazon EC2 para utilizarla como cliente para 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. Cree una instancia de Amazon EC2 para utilizarla como un equipo 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 tenga AWS CLI instalado, ejecute el siguiente comando para obtener los agentes de arranque del clúster. *Cluster-ARN* Sústítúyalo 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 marcadores de posición por su versión de cliente de Kafka. *{YOUR KAFKA VERSION}*

```
security.protocol=SSL
ssl.truststore.location=/tmp/kafka_2.12-{YOUR KAFKA VERSION}/
kafka.client.truststore.jks
ssl.keystore.location=/tmp/kafka_2.12-{YOUR KAFKA VERSION}/
kafka.client.keystore.jks
ssl.keystore.password=Your-Store-Pass
```

```
ssl.key.password=Your-Key-Pass
```

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.

Sign-in autenticación de credenciales 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.

Después de asociar un secreto a un clúster de MSK, MSK sincroniza periódicamente los datos de credenciales.

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

Sign-in la autenticación de credenciales 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 el 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 “Dimensione correctamente el clúster: número de particiones por agente Standard”](#).

- 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 [BatchAssociateScramSecret](#) operación.

⚠ 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 cambios en la política del recurso de Secretos o en la clave de KMS utilizada para el cifrado del secreto, asegúrese de volver a asociar los secretos al clúster de MSK. Esto garantiza que el clúster mantenga el acceso al 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 datos desde un tema de ejemplo.

Temas

- [Conectar un cliente a un clúster mediante SASL/SCRAM autenticación](#)
- [Solución de problemas de conexión](#)

Conectar 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
```

Del resultado JSON de este comando, guarde el valor asociado a la cadena llamada `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 del JDK desde la carpeta `cacerts` del JVM al archivo `kafka.client.truststore.jks` 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, ejecute el siguiente comando.
BootstrapBrokerStringSaslScram Sustitúyala por la cadena del broker bootstrap 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

Cuando ejecute comandos de cliente de Kafka, es posible que encuentre errores de memoria del montón de Java, especialmente al trabajar con temas o conjuntos de datos de gran tamaño. Estos errores se producen porque las herramientas de Kafka se ejecutan como aplicaciones Java con configuraciones de memoria predeterminadas que pueden ser insuficientes para la carga de trabajo.

Para resolver los errores Out of Memory Java Heap, puede aumentar el tamaño del montón de Java mediante la modificación de la variable de entorno KAFKA_OPTS para incluir la configuración de memoria.

El siguiente ejemplo 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"
```

Para consumir temas de gran tamaño, considere el uso de parámetros basados en tiempo o en desplazamientos, en lugar de --from-beginning, con el fin de limitar el uso de memoria.

```
<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 [ACL de Apache Kafka](#).

En el caso de los clústeres que utilizan el ZooKeeper modo, se recomienda restringir el acceso a ZooKeeper los nodos para evitar que los usuarios modifiquen las ACL. 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 SCRAM-SHA-512 autenticación.
- 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 [BatchAssociateScramSecret](#) operación.
- 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.

ACL de Apache Kafka

Apache Kafka tiene un autorizador conectable y se envía con una implementación de autorizador lista para usar. Amazon MSK habilita este autorizador en el archivo `server.properties` de los agentes.

Las ACL de Apache Kafka tienen el formato «La P principal es [Allowed/Denied] la operación O del host H en cualquier recurso R que coincida con el ResourcePattern RP». Si RP no coincide con un recurso específico R, R no tiene ACL asociadas y, por lo tanto, nadie más que los superusuarios tiene permiso para acceder a R. Para cambiar este comportamiento de Apache Kafka, establezca la propiedad `allow.everyone.if.no.acl.found` en `true`. Amazon MSK lo establece como `true` de forma predeterminada. Esto significa que con los clústeres de Amazon MSK, si no establece explícitamente las ACL en un recurso, todos los principales pueden acceder a este recurso. Si habilita las ACL en un recurso, sólo los principales autorizados pueden acceder a él. Si desea restringir el acceso a un tema y autorizar a un cliente mediante la autenticación mutua de TLS, agregue ACL mediante la CLI del autorizador de Apache Kafka. Para obtener más información acerca de cómo agregar, eliminar y enumerar ACL, consulte la [Interfaz de línea de comandos de Kafka](#).

Dado que Amazon MSK configura los agentes como superusuarios, estos pueden acceder a todos los temas. Esto ayuda a los agentes a replicar mensajes desde la partición principal, independientemente de que la propiedad `allow.everyone.if.no.acl.found` esté definida o no en la configuración del clúster.

Adición o eliminación del acceso de lectura y escritura a un tema

1. Agregue sus agentes a la tabla de ACL para permitirles leer de todos los temas que tengan ACL en su lugar. 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.

Distinguished-Name Sustitúyalo 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 DNS `b-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 “Obtención de los agentes de arranque”](#).

```
<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 la máquina 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 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 los ID de las interfaces de red elásticas (ENI) que están asociadas a los agentes.

2. Inicie sesión en la consola Amazon EC2 Consola de administración de AWS y ábrala 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. Repita este paso para cada una de las ENI que obtuvo en el primer paso.

 Note

Los cambios que realice en el grupo de seguridad de un clúster mediante la consola de Amazon EC2 no se reflejan en la consola de MSK en Configuración de redes.

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 agrega nuevos agentes después de cambiar el grupo de seguridad, debe volver a seguir el procedimiento anterior y actualizar las ENI de los nuevos agentes.

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 el modo KRaft. Consulte [the section called “Modo KRaft”](#).

Este tema contiene las siguientes secciones:

- [Habilite o deshabilite el acceso directo a los ZooKeeper clientes de Apache](#)
- [Para colocar ZooKeeper los nodos de Apache en un grupo de seguridad independiente](#)
- [Uso de la seguridad TLS con Apache ZooKeeper](#)

Habilite o deshabilite el acceso directo a los ZooKeeper clientes de Apache

Puede deshabilitar el acceso directo de los ZooKeeper clientes Apache en su clúster aprovisionado de Amazon MSK para comprobar que sus aplicaciones no dependen de conexiones directas ZooKeeper . Cuando el ZooKeeper acceso está deshabilitado, los clientes ya no pueden conectarse a ZooKeeper los nodos de Apache en los puertos 2181 (texto sin formato) y 2182 (TLS). Puede volver a habilitar ZooKeeper el acceso en cualquier momento.

 Note

Esta función solo está disponible para los clústeres aprovisionados de Amazon MSK que utilizan el modo de ZooKeeper metadatos con agentes estándar. No está disponible para los siguientes tipos de clústeres:

- Clústeres que se ejecutan en el modo de metadatos de KRaft
- Clústeres que utilizan corredores Express. ZooKeeper el acceso se administra automáticamente en los clústeres de Express y no se puede configurar manualmente.
- Clústeres Amazon MSK Serverless

Inhabilitar el ZooKeeper acceso mediante la consola

1. [Inicie sesión en y abra la Consola de administración de AWS consola de Amazon MSK en? https://console.aws.amazon.com/msk/home region=us-east-1#/home/](https://console.aws.amazon.com/msk/home region=us-east-1#/home/).
2. En la lista de clústeres, elija el clúster en el que desee deshabilitar el acceso. ZooKeeper
3. Seleccione la pestaña Propiedades y, a continuación, busque la sección Configuración de redes.
4. Seleccione Desactivar el ZooKeeper acceso.

Inhabilitar el ZooKeeper acceso mediante el AWS CLI

1. Ejecute el siguiente AWS CLI comando, sustituyendo *ClusterArn* y *Current-Cluster-Version* por el ARN y la versión actual del clúster. Para encontrar la versión actual del clúster, utilice la [DescribeCluster](#) operación o el comando [AWS CLI describe-cluster](#). Un ejemplo de ID de versión es KTVDPKIKX0DER.

```
aws kafka update-connectivity --cluster-arn ClusterArn --current-version Current-Cluster-Version --zookeeper-access '{"Enabled": false}'
```

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 volver a habilitar el ZooKeeper acceso, utilice un AWS CLI comando similar con el siguiente valor for en su lugar: --zookeeper-access

```
'{"Enabled": true}'
```

2. Para obtener el resultado de la update-connectivity operación, ejecute el siguiente comando y *ClusterOperationArn* reemplácelo por el ARN que obtuvo en el resultado 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": "2026-01-15T21:08:57.735Z",
    "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": {
      "ZookeeperAccess": {
        "Enabled": true
      }
    },
    "TargetClusterInfo": {
      "ZookeeperAccess": {
        "Enabled": false
      }
    }
  }
}
```

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

Inhabilitar el ZooKeeper acceso mediante la API de Amazon MSK

- Para usar la API para deshabilitar o habilitar el ZooKeeper acceso a un clúster, consulte. [UpdateConnectivity](#)

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. Inicie sesión en la consola Amazon EC2 Consola de administración de AWS y ábrala 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 en la parte inferior de la página, busque la IP IPv4 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 [KIP-515: Habilitar el cliente ZK para que utilice la nueva autenticación compatible con TLS](#).

Validación de la conformidad de Amazon Managed Streaming para Apache Kafka

Third-party los auditores evalúan la seguridad y la conformidad de Amazon Managed Streaming for Apache Kafka como parte de los programas de AWS conformidad. 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: este documento técnico](#) describe cómo pueden utilizar las empresas para crear aplicaciones. AWS HIPAA-compliant
- [AWS Recursos de cumplimiento Recursos AWS](#) : esta colección de libros 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 CSPM](#)— 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 admitir conjuntos de cifrado con perfecto secreto directo (PFS), como Ephemeral (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Diffie-Hellman 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.

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 están disponibles tanto en los corredores MSK Standard como en los 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 los registros de los INFO-level corredores 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: AmazonMSKFullAccess](#).

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 obligatoria para su uso con SSE-KMS buckets

Si ha activado el cifrado del lado del servidor para su bucket de S3 mediante AWS KMS-managed keys (SSE-KMS) con una clave gestionada 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 Consola de administración de AWS

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.

 Note

Debe establecer la etiqueta `LogDeliveryEnabled` en `true` en los flujos de Firehose para configurar la entrega de registros. El rol vinculado al servicio que AWS crea CloudWatch los registros usa esta etiqueta para conceder permisos a todas las transmisiones de entrega de Firehose. Si elimina esta etiqueta, el rol vinculado al servicio no podrá entregar registros al flujo de Firehose. Para ver un ejemplo de una política de IAM que muestre los permisos que incluye la función vinculada al servicio, consulte [Funciones de IAM utilizadas para los permisos de recursos en](#) la Guía del usuario de Amazon. CloudWatch

```
{
  "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

Puedes especificar la `loggingInfo` estructura opcional en el JSON que vas a transferir a las operaciones o. [CreateClusterUpdateMonitoring](#)

Note

De forma predeterminada, cuando el registro de agentes está habilitado, Amazon MSK almacena los registros de INFO en los destinos especificados. Sin embargo, en el caso de los corredores estándar, los usuarios de Apache Kafka 2.4.X y versiones posteriores pueden establecer dinámicamente el nivel de registro del corredor en cualquiera de los niveles de registro de [log4j](#). Para obtener información sobre cómo configurar dinámicamente el nivel de registro del corredor, consulte [KIP-412: Ampliar la API de administración para admitir los niveles de registro dinámicos de las aplicaciones](#). 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.

Registrar llamadas a la API con AWS CloudTrail

Note

AWS CloudTrail los registros están disponibles para Amazon MSK solo cuando los usas [Control 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": "ABCDEFGH0123456789ABCDE",
    "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 metadatos de Apache ZooKeeper o KrAFT.

A partir de la versión 3.7.x de Apache Kafka en Amazon MSK, puede crear clústeres que utilicen el modo KrAFT en lugar del modo. ZooKeeper KRaft-based. Los clústeres dependen de los controladores de Kafka para administrar los metadatos.

Temas

- [ZooKeeper modo](#)
- [Modo KRaft](#)

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 Consola de administración de AWS

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.

Obtener la cadena de ZooKeeper conexión de Apache mediante el AWS CLI

1. Si no conoce 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* sustitúyalo 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](#).

Modo KRaft

Amazon MSK introdujo la compatibilidad con KRaft (Apache Kafka Raft) en la versión 3.7.x de Kafka. La comunidad de Apache Kafka desarrolló KrAFT para sustituir a [Apache ZooKeeper](#) en la gestión de metadatos en los clústeres de Apache Kafka. En el modo KrAFT, 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 Los controladores KRaft se incluyen sin costo adicional para usted y no requieren ninguna configuración o administración adicionales por su parte. Consulte [KIP-500](#) para obtener más información sobre KrAFT.

Estos son algunos puntos a tener en cuenta sobre el modo KRaft en MSK:

- El modo KRaft 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 KRaft-based clúster seleccionando la versión 3.7.x de Kafka y marcando la casilla de verificación KRaft en la ventana de creación del clúster.
- Para crear un clúster en modo KRaft mediante la API [CreateCluster](#) de MSK u operaciones [CreateClusterV2](#), debe usar 3.7.x.kraft como versión. 3.7.xUtilícela como versión para crear un clúster en modo. ZooKeeper
- El número de particiones por agente es el mismo en los clústeres de KRaft y en los clústeres ZooKeeper basados. 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 utilizar el modo KRaft 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 para todas las versiones publicadas en las que Zookeeper también es compatible con Apache Kafka. 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 puedan utilizar las API de administración de Kafka sin conexiones. 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 [ejecutar el Cruise Control sin ZooKeeper ella](#).
- No necesita acceder directamente a los controladores KRaft de su clúster para realizar alguna 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. Puedes obtener estos puntos finales de DNS desde la consola MSK o mediante la operación de la [ListNodesAPI](#). Consulte los pasos actualizados [Supervisión de un clúster de MSK aprovisionado con Prometheus](#) para configurar la supervisión abierta de los clústeres. KRaft-based
- No hay [CloudWatch métricas](#) adicionales que debas monitorizar para los clústeres en modo KRaft sobre los clústeres en ZooKeeper modo. MSK administra los controladores KRaft que se utilizan en sus clústeres.

- Puede seguir administrando las ACL utilizando clústeres en modo KRaft mediante la cadena de conexión `--bootstrap-server`. No debe utilizar la cadena de conexión `--zookeeper` para administrar las ACL. Consulte [ACL de Apache Kafka](#).
- En el modo KRaft, los metadatos del clúster se almacenan en los controladores KRaft de Kafka y no en nodos externos. ZooKeeper Por lo tanto, no necesita controlar el acceso a los nodos del controlador por separado [como lo hace](#) con los nodos. ZooKeeper

Tema: Operaciones

Puede usar Amazon MSK APIs para administrar los temas de su clúster aprovisionado de MSK sin necesidad de configurar y mantener los clientes de administración de Kafka. Con ellos APIs, puede definir o leer las propiedades de los temas, como el factor de replicación y el recuento de particiones, junto con los ajustes de configuración, como las políticas de retención y limpieza. Puede administrar los temas de Kafka mediante programación mediante sus interfaces conocidas, incluidas AWS CLI, y AWS SDKs. AWS CloudFormation También APIs están integradas en la consola de Amazon MSK, lo que permite agrupar todas las operaciones relacionadas con los temas en un solo lugar. Ahora puede crear o actualizar temas con solo unos pocos clics utilizando los valores predeterminados guiados y, al mismo tiempo, obtener una visibilidad completa de las configuraciones de los temas, la información a nivel de partición y las métricas.

Important

Las respuestas de la API de estos temas reflejan datos que se actualizan aproximadamente cada minuto. Para ver el estado más actual del tema después de realizar cambios, espere aproximadamente un minuto antes de realizar la consulta.

Requisitos para usar el tema APIs

- El clúster debe ser un clúster aprovisionado por MSK. No APIs están disponibles para los clústeres sin servidor de MSK.
- El clúster debe ejecutar la versión 3.6.0 o posterior de Apache Kafka. Para obtener más información sobre las versiones compatibles, consulte [the section called “Versiones compatibles de Apache Kafka”](#)
- El clúster debe estar en ese ACTIVE estado. Para obtener más información acerca de los estados del clúster, consulte [Descripción de los estados de los clústeres de MSK aprovisionados](#).

- Debe tener los permisos de IAM adecuados. Para obtener más información, consulte [the section called “Permisos de IAM”](#).

Permisos de IAM para las operaciones de los temas APIs

Para llamarlas APIs, debe tener los permisos de IAM adecuados. En la siguiente tabla se enumeran los permisos necesarios para cada API.

Permisos necesarios para las operaciones de los temas APIs

API	Permisos necesarios	Recurso
ListTopics	kafka-cluster:Connect kafka-cluster:DescribeTopic	Clúster: ARN, tema: ARN
DescribeTopic	kafka-cluster:Connect kafka-cluster:DescribeTopic kafka-cluster:DescribeTopic DynamicConfiguration	Clúster: ARN, tema: ARN
DescribeTopicPartitions	kafka-cluster:Connect kafka-cluster:DescribeTopic kafka-cluster:DescribeTopic DynamicConfiguration	Clúster: ARN, tema: ARN
CreateTopic	kafka-cluster:Connect kafka-cluster>CreateTopic	Clúster: ARN, tema: ARN
DeleteTopic	kafka-cluster:Connect kafka-cluster:DescribeTopic kafka-cluster>DeleteTopic	Clúster: ARN, tema: ARN
UpdateTopic	kafka-cluster:Connect	Clúster: ARN, tema: ARN

API	Permisos necesarios	Recurso
	<code>kafka-cluster:DescribeTopic</code> <code>kafka-cluster:AlterTopic</code> <code>kafka-cluster:AlterTopicDynamicConfiguration</code>	

Note

Para `kafka-cluster:Connect`, especifique el ARN del clúster en su política de IAM. Para todas las demás acciones, especifique el ARN del tema en su política de IAM.

Note

Para `elloListTopics`, puede utilizar un comodín (*) para hacer coincidir todos los temas de un clúster. Por ejemplo: `arn:aws:kafka:us-east-1:123456789012:topic/my-cluster/abcd1234-abcd-dcba-4321-a1b2abcd9f9f-2/*`.

Para obtener más información sobre el control de acceso de IAM para Amazon MSK, consulte. [the section called “Control de acceso de IAM”](#)

Temas

- [Listar los temas de un clúster de Amazon MSK](#)
- [Obtenga información detallada sobre un tema](#)
- [Ver la información de la partición de un tema](#)
- [Crear temas en un clúster de Amazon MSK](#)
- [Actualizar un tema en un clúster de Amazon MSK](#)
- [Eliminar un tema de un clúster de Amazon MSK](#)

Listar los temas de un clúster de Amazon MSK

Puede enumerar todos los temas de su clúster aprovisionado de MSK para ver los metadatos básicos, como el número de particiones y los factores de replicación. Esto resulta útil para supervisar los temas del clúster, realizar comprobaciones de inventario o identificar los temas que deben investigarse más a fondo.

Note

La `ListTopics` API proporciona metadatos de temas básicos. Para obtener información detallada sobre un tema específico, incluidos su estado y configuración actuales, usa la `DescribeTopic` API. Para obtener más información, consulte [Obtenga información detallada sobre un tema](#).

Note

Esta respuesta de la API refleja los datos que se actualizan aproximadamente cada minuto. Para ver el estado más actual del tema después de realizar cambios, espere aproximadamente un minuto antes de realizar la consulta.

Temas

- [Enumere los temas mediante el Consola de administración de AWS](#)
- [Enumere los temas utilizando el AWS CLI](#)
- [Enumere los temas que utilizan la API](#)

Enumere los temas mediante el Consola de administración de AWS

1. ¿Iniciar sesión en la Consola de administración de AWS 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 el nombre del clúster para el que desee enumerar los temas.
3. En la página de detalles del clúster, seleccione la pestaña Temas.
4. En la tabla se muestran todos los temas del clúster, incluidos el nombre del tema, el número de particiones, el factor de replicación y el recuento de out-of-sync réplicas.

Enumere los temas utilizando el AWS CLI

Ejecute el siguiente comando y *ClusterArn* sustitúyalo por el nombre de recurso de Amazon (ARN) del 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 list-topics --cluster-arn ClusterArn
```

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

```
{
  "topics": [
    {
      "topicArn": "arn:aws:kafka:us-east-1:123456789012:topic/MyCluster/abcd1234-
abcd-dcba-4321-a1b2abcd9f9f-2/MyTopic",
      "topicName": "MyTopic",
      "partitionCount": 3,
      "replicationFactor": 3,
      "outOfSyncReplicaCount": 0
    },
    {
      "topicArn": "arn:aws:kafka:us-east-1:123456789012:topic/MyCluster/abcd1234-
abcd-dcba-4321-a1b2abcd9f9f-2/AnotherTopic",
      "topicName": "AnotherTopic",
      "partitionCount": 6,
      "replicationFactor": 3,
      "outOfSyncReplicaCount": 1
    }
  ]
}
```

Paginación de resultados

Si su clúster tiene muchos temas, puede usar la paginación para recuperar los resultados en lotes más pequeños. Utilice el `--max-results` parámetro para especificar el número máximo de temas que se van a devolver y utilice el `--next-token` parámetro para recuperar la siguiente página de resultados.

```
aws kafka list-topics --cluster-arn ClusterArn --max-results 10
```


Si hay más resultados disponibles, la respuesta incluye un `nextToken` valor. Utilice este token para recuperar la siguiente página de resultados.

```
aws kafka list-topics --cluster-arn ClusterArn --max-results 10 --next-token NextToken
```

Filtrar temas por nombre

Puede filtrar la lista de temas especificando un prefijo mediante el `--topic-name-filter` parámetro. Esto devuelve solo los temas cuyos nombres comiencen con el prefijo especificado.

```
aws kafka list-topics --cluster-arn ClusterArn --topic-name-filter "prod-"
```

Este comando devuelve solo los temas cuyos nombres comiencen por `prod-`, como `prod-orders` o `prod-inventory`.

Enumere los temas que utilizan la API

Para ver una lista de los temas que utilizan la API, consulte [ListTopics](#).

Obtenga información detallada sobre un tema

Puede recuperar información detallada sobre un tema específico de su clúster provisionado de MSK, incluido su estado actual, el número de particiones, el factor de replicación y la configuración. Esto resulta útil para solucionar problemas, validar la configuración de los temas o supervisar el estado de los temas durante las operaciones.

Note

Esta respuesta de la API refleja los datos que se actualizan aproximadamente cada minuto. Para ver el estado más actual del tema después de realizar cambios, espere aproximadamente un minuto antes de realizar la consulta.

Temas

- [Describa un tema utilizando el Consola de administración de AWS](#)
- [Describa un tema utilizando el AWS CLI](#)
- [Describa un tema mediante la API](#)

Describa un tema utilizando el Consola de administración de AWS

1. ¿Iniciar sesión en la Consola de administración de AWS 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 el nombre del clúster que contiene el tema que desea describir.
3. En la página de detalles del clúster, seleccione la pestaña Temas.
4. En la lista de temas, elige el nombre del tema que quieres ver.
5. La página de detalles del tema muestra información sobre el tema, incluidos su estado, el número de particiones, el factor de replicación y los ajustes de configuración.

Describa un tema utilizando el AWS CLI

Ejecute el siguiente comando y *ClusterArn* sustitúyalo por el nombre de recurso de Amazon (ARN) del clúster y *TopicName* por el nombre del tema que desee describir.

```
aws kafka describe-topic --cluster-arn ClusterArn --topic-name TopicName
```

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

```
{
  "topicArn": "arn:aws:kafka:us-east-1:123456789012:topic/MyCluster/abcd1234-abcd-dcba-4321-a1b2abcd9f9f-2/MyTopic",
  "topicName": "MyTopic",
  "partitionCount": 3,
  "replicationFactor": 3,
  "configs": "Y29tcHJlc3Npb24udHlwZT1wcm9kdWNlcgpyZXR1bnRpb24ubXM9NjA0ODAwMDAw",
  "status": "ACTIVE"
}
```

Entender el estado del tema

El `status` campo indica el estado actual del tema. En la siguiente tabla se describen los posibles valores de estado.

Valores de estado del tema

Status	Description (Descripción)
CREATING	Se está creando el tema.

Status	Description (Descripción)
ACTIVE	El tema está activo y listo para usarse.
ACTUALIZANDO	Se está actualizando la configuración del tema.
ELIMINANDO	Se está eliminando el tema.

Descripción de las configuraciones de los temas

El `configs` campo contiene las propiedades de configuración de Kafka del tema, codificadas en formato Base64. Para ver la configuración en un formato legible, debe decodificar la cadena Base64.

El siguiente ejemplo muestra cómo decodificar la configuración mediante el `base64` comando en Linux o macOS.

```
echo "Y29tcHJlc3Npb24udHlwZT1wcm9kdWNlcgpyZXR1bnRpb24ubXM9NjA0ODAwMDAw" | base64 --  
decode
```

El resultado decodificado muestra las propiedades de configuración del tema en formato clave-valor.

```
compression.type=producer  
retention.ms=604800000
```

Para obtener más información sobre las propiedades de configuración a nivel de tema, consulte [the section called “Topic-level Configuración de Amazon MSK”](#)

Describe un tema mediante la API

Para describir un tema sobre el uso de la API, consulte [DescribeTopic](#).

Ver la información de la partición de un tema

Puedes recuperar información detallada sobre las particiones de un tema específico de tu clúster provisionado de MSK. Esta información incluye el número de partición, el agente líder, los agentes de réplicas y las réplicas sincronizadas (ISR). Esto resulta útil para supervisar la distribución de las particiones, identificar las particiones que no están suficientemente replicadas o solucionar problemas de replicación.

 Note

Esta respuesta de la API refleja los datos que se actualizan aproximadamente cada minuto. Para ver el estado más actual del tema después de realizar cambios, espere aproximadamente un minuto antes de realizar la consulta.

Temas

- [Vea la información de la partición mediante el Consola de administración de AWS](#)
- [Vea la información de la partición mediante el AWS CLI](#)
- [Vea la información de las particiones mediante la API](#)

Vea la información de la partición mediante el Consola de administración de AWS

1. ¿Iniciar sesión en la Consola de administración de AWS 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 el nombre del clúster que contiene el tema.
3. En la página de detalles del clúster, seleccione la pestaña Temas.
4. En la lista de temas, elija el nombre del tema del que quiere ver la información de la partición.
5. En la página de detalles del tema, se muestra la información de la partición, con el número de partición, el agente principal, las réplicas y las réplicas sincronizadas de cada partición.

Vea la información de la partición mediante el AWS CLI

Ejecute el siguiente comando y *ClusterArn* sustitúyalo por el nombre de recurso de Amazon (ARN) del clúster y *TopicName* por el nombre del tema.

```
aws kafka describe-topic-partitions --cluster-arn ClusterArn --topic-name TopicName
```

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

```
{
  "partitions": [
    {
      "partition": 0,
      "leader": 1,
```

```
    "replicas": [1, 2, 3],
    "isr": [1, 2, 3]
  },
  {
    "partition": 1,
    "leader": 2,
    "replicas": [2, 3, 1],
    "isr": [2, 3, 1]
  },
  {
    "partition": 2,
    "leader": 3,
    "replicas": [3, 1, 2],
    "isr": [3, 1]
  }
]
```

Entender la información de las particiones

La respuesta incluye la siguiente información para cada partición:

- **partición**: el número de partición. Las particiones se numeran empezando por 0.
- **líder**: el ID de intermediario del líder de esta partición. El líder gestiona todas las solicitudes de lectura y escritura de la partición.
- **réplicas**: la lista de corredores IDs que tienen réplicas de esta partición. Esto incluye tanto las réplicas como las sincronizadas. out-of-sync
- **isr**: la lista de corredores IDs que son réplicas sincronizadas. Estas réplicas están totalmente alineadas con el líder y, si es necesario, pueden asumir el papel de líder.

En el ejemplo anterior, la partición 2 tiene una out-of-sync réplica. La `replicas` lista incluye al corredor 2, pero no. `isr` Esto indica que el corredor 2 no está completamente al mismo nivel que el líder de esta partición.

Paginación de resultados

Si el tema tiene muchas particiones, puede usar la paginación para recuperar los resultados en lotes más pequeños. Utilice el `--max-results` parámetro para especificar el número máximo de particiones que se van a devolver y utilice el `--next-token` parámetro para recuperar la siguiente página de resultados.

```
aws kafka describe-topic-partitions --cluster-arn ClusterArn --topic-name TopicName --  
max-results 10
```

Si hay más resultados disponibles, la respuesta incluye un `nextToken` valor. Utilice este token para recuperar la siguiente página de resultados.

```
aws kafka describe-topic-partitions --cluster-arn ClusterArn --topic-name TopicName --  
max-results 10 --next-token NextToken
```

Casos de uso comunes

La visualización de la información de las particiones es útil en varios escenarios:

- Identificación de las particiones infrarreplicadas: compare las `isr` listas `replicas` y para identificar las particiones en las que algunas réplicas no están sincronizadas. Esto puede indicar problemas de rendimiento o problemas con los intermediarios.
- Supervisión de la distribución de las particiones: compruebe que los líderes de las particiones estén distribuidos uniformemente entre los corredores para garantizar una carga equilibrada.
- Solución de problemas de replicación: examine la lista de ISR para identificar qué intermediarios tienen problemas para mantener el ritmo de la replicación.
- Planear el reequilibrio de particiones: utilice esta información para comprender el diseño actual de las particiones antes de realizar las operaciones de reequilibrio.

Vea la información de las particiones mediante la API

Para ver la información de las particiones mediante la API, consulte [DescribeTopicPartitions](#).

Crear temas en un clúster de Amazon MSK

Puedes crear temas en tu clúster aprovisionado de MSK directamente con esta API sin tener que configurar ningún Kafka personalizado. `AdminClient` Al crear un tema, debe especificar el nombre del tema, el número de particiones, el factor de replicación y, si lo desea, las configuraciones del tema.

Temas

- [Cree temas mediante el Consola de administración de AWS](#)
- [Cree un tema mediante el AWS CLI](#)
- [Cree un tema mediante la API](#)

Cree temas mediante el Consola de administración de AWS

1. ¿Iniciar sesión en la Consola de administración de AWS 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 el nombre del clúster en el que desee crear los temas.
3. En la página de detalles del clúster, seleccione la pestaña Temas.
4. Seleccione Crear tema.
5. Introduzca el nombre del tema, el número de particiones y el factor de replicación. Si lo desea, añada configuraciones. Puede crear varios temas a la vez.
6. Seleccione Crear tema.

Cree un tema mediante el AWS CLI

Ejecute el siguiente comando y *ClusterArn* sustitúyalo por el nombre de recurso de Amazon (ARN) del 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 create-topic --cluster-arn ClusterArn --topic-name MyTopic --partition-count 3 --replication-factor 3
```

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

```
{
  "topicArn": "arn:aws:kafka:us-east-1:123456789012:topic/MyCluster/abcd1234-abcd-dcba-4321-a1b2abcd9f9f-2/MyTopic",
  "topicName": "MyTopic",
  "status": "CREATING"
}
```

Cree un tema mediante la API

Para crear un tema con la API, consulte [CreateTopic](#).

Actualizar un tema en un clúster de Amazon MSK

Actualice el recuento de particiones o las configuraciones a nivel de tema de un tema existente. Esta operación modifica el tema sin necesidad de volver a crearlo.

 Note

Puede actualizar el recuento de particiones o las configuraciones de los temas en una sola llamada a la API, pero no de forma simultánea. Para actualizar ambas, realiza llamadas a la API por separado.

Temas

- [Actualiza un tema mediante el Consola de administración de AWS](#)
- [Actualice un tema mediante el AWS CLI](#)
- [Actualiza un tema mediante la API](#)

Actualiza un tema mediante el Consola de administración de AWS

1. ¿Iniciar sesión en la Consola de administración de AWS 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 el nombre del clúster que contiene el tema que desee actualizar.
3. En la página de detalles del clúster, seleccione la pestaña Temas.
4. Seleccione el tema que desee actualizar y, a continuación, elija Editar la configuración de la partición o Editar la configuración desde Acciones.
5. Actualice el recuento de particiones o las configuraciones según sea necesario.
6. Seleccione Save.

Actualice un tema mediante el AWS CLI

Ejecuta el siguiente comando y *ClusterArn* sustitúyelo por el nombre de recurso de Amazon (ARN) del clúster y *TopicName* por el nombre del tema que desees actualizar.

```
aws kafka update-topic --cluster-arn ClusterArn --topic-name TopicName --partition-count 6
```

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

```
{
  "topicArn": "arn:aws:kafka:us-east-1:123456789012:topic/MyCluster/abcd1234-abcd-dcba-4321-a1b2abcd9f9f-2/MyTopic",
```



```
"topicName": "MyTopic",  
"status": "UPDATING"  
}
```

Actualiza un tema mediante la API

Para actualizar un tema mediante la API, consulte [UpdateTopic](#).

Eliminar un tema de un clúster de Amazon MSK

Al eliminar un tema, se eliminan permanentemente todos sus datos, metadatos e información de particiones. Esta operación no se puede deshacer.

Temas

- [Elimine un tema mediante el Consola de administración de AWS](#)
- [Elimine un tema mediante el AWS CLI](#)
- [Elimine un tema mediante la API](#)

Elimine un tema mediante el Consola de administración de AWS

1. ¿Iniciar sesión en la Consola de administración de AWS 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 el nombre del clúster que contiene el tema que desea eliminar.
3. En la página de detalles del clúster, seleccione la pestaña Temas.
4. Seleccione los temas que desee eliminar y, a continuación, elija Eliminar de las acciones.
5. Confirme la eliminación y, a continuación, seleccione Eliminar.

Elimine un tema mediante el AWS CLI

Ejecuta el siguiente comando y *ClusterArn* sustitúyelo por el nombre de recurso de Amazon (ARN) del clúster y *TopicName* por el nombre del tema que deseas eliminar.

```
aws kafka delete-topic --cluster-arn ClusterArn --topic-name TopicName
```

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

```
{
```

```
"topicArn": "arn:aws:kafka:us-east-1:123456789012:topic/MyCluster/abcd1234-abcd-dcba-4321-a1b2abcd9f9f-2/MyTopic",
"topicName": "MyTopic",
"status": "DELETING"
}
```

Elimine un tema mediante la API

Para eliminar un tema mediante la API, consulte [DeleteTopic](#).

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 información, consulte la [política de fin de vida \(EOL\) de Apache Kafka](#).

La siguiente tabla enumera las versiones de Apache Kafka que admite Amazon MSK.

Versión Apache Kafka	Fecha de publicación de MSK	Fecha de finalización del soporte
1.1.1	--	05 de junio de 2024
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
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

Versión Apache Kafka	Fecha de publicación de MSK	Fecha de finalización del soporte
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	2026-06-01
3.7.x	2024-05-29	2026-09-01
3.8.x	2025-02-20	--
3.9.x (recomendado)	2025-04-21	--
4.0.x	2025-05-16	--
4.1.x	15-10-25	--

Para obtener más información sobre la política de compatibilidad de versiones de Amazon MSK, consulte [Política de soporte de las versiones de Amazon MSK](#).

Amazon MSK versión 4.1.x

Amazon Managed Streaming para Apache Kafka (Amazon MSK) ahora admite Apache Kafka versión 4.1, que introduce Colas como una característica en versión preliminar, un nuevo Protocolo de reequilibrio de Flujos con acceso anticipado y Réplicas líderes elegibles (ELR). Junto con estas características, Apache Kafka versión 4.1 incluye diversas correcciones de errores y mejoras.

Un aspecto destacado clave de Kafka 4.1 es la introducción de Colas como característica en versión preliminar. Puede usar varios consumidores para procesar mensajes desde las mismas particiones de temas, lo que mejora el paralelismo y el rendimiento para cargas de trabajo que requieren entrega de mensajes punto a punto. El nuevo Protocolo de reequilibrio de Flujos se basa en el protocolo de reequilibrio de consumidores de Kafka 4.0 y amplía las capacidades de coordinación del agente para Kafka Streams, lo que permite asignaciones de tareas optimizadas y un reequilibrio más eficiente. Además, ELR ahora está habilitado de forma predeterminada para reforzar la disponibilidad.

Para obtener más información y una lista completa de mejoras y correcciones de errores, consulte las [notas de la versión de Apache Kafka 4.1](#).

Para empezar a usar Apache Kafka 4.1 en Amazon MSK, elija la versión 4.1.x al crear un nuevo clúster a través de la Consola de administración de AWS, AWS CLI o los SDK. AWS También puede actualizar clústeres de MSK aprovisionados existentes mediante una actualización gradual en el lugar. Amazon MSK orquesta los reinicios de los agentes para mantener la disponibilidad y proteger los datos durante la actualización. La compatibilidad con la versión 4.1 de Kafka está disponible en todos los Regiones de AWS sitios donde se ofrece Amazon MSK.

Amazon MSK versión 4.0.x

Amazon Managed Streaming para Apache Kafka (Amazon MSK) ahora admite Apache Kafka versión 4.0. Esta versión incorpora los avances más recientes en administración de clústeres y rendimiento en MSK aprovisionado. Kafka 4.0 introduce un nuevo protocolo de reequilibrio de consumidores, ahora disponible de forma general, que ayuda a garantizar reequilibrios de grupos más fluidos y rápidos. Además, Kafka 4.0 requiere que los intermediarios y las herramientas utilicen Java 17, lo que proporciona una seguridad y un rendimiento mejorados, incluye varias correcciones de errores y mejoras, y hace obsoleta la administración de metadatos a través de Apache. ZooKeeper

Para obtener más información y una lista completa de mejoras y correcciones de errores, consulte las [notas de la versión de Apache Kafka 4.0](#).

Amazon MSK versión 3.9.x (recomendada)

Amazon Managed Streaming para Apache Kafka (Amazon MSK) ahora admite Apache Kafka versión 3.9. Esta versión mejora la funcionalidad de almacenamiento por niveles, ya que permite conservar los datos almacenados por niveles cuando se desactiva el almacenamiento por niveles en los temas. Las aplicaciones consumidoras pueden leer datos históricos desde el desplazamiento inicial del registro remoto (Rx), al tiempo que se mantienen desplazamientos de registro continuos entre el almacenamiento local y el remoto.

La versión 3.9 es la última versión compatible con los sistemas de administración de metadatos de ZooKeeper KrAFT. Amazon MSK proporcionará compatibilidad ampliada para la versión 3.9 durante un período mínimo de dos años a partir de su fecha de lanzamiento.

Para obtener más información y una lista completa de mejoras y correcciones de errores, consulte las [notas de la versión de Apache Kafka 3.9](#).

Amazon MSK versión 3.8.x

Amazon Managed Streaming para Apache Kafka (Amazon MSK) ahora admite Apache Kafka versión 3.8. 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. Apache Kafka versión 3.8 incluye varias correcciones de errores y nuevas características que mejoran el rendimiento. Entre las principales características nuevas se incluye la compatibilidad con la configuración del nivel de compresión. Esta versión permite optimizar aún más el rendimiento al usar 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 de Apache Kafka 3.8.x](#).

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.

Ahora, con la versión 3.7.x de Apache Kafka, Amazon MSK es compatible con el modo KRaft (Apache Kafka Raft). En Amazon MSK, al igual que con ZooKeeper los nodos, los controladores KRAFT se incluyen sin coste adicional para usted y no requieren ninguna configuración o administración adicionales por su parte. Ahora puede crear clústeres en modo KRAFT o en modo Apache Kafka ZooKeeper versión 3.7.x. En el modo Kraft, puede añadir hasta 60 agentes para alojar más particiones por clúster, sin solicitar un aumento del límite, en comparación con la cuota de 30 agentes de clústeres. Zookeeper-based Para obtener más información sobre KRaft en Amazon MSK, consulte [Modo KRaft](#).

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. Entre las características más importantes, se incluyen mejoras en las métricas y en el uso de los ID de tema. 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 de Amazon de la MSK-only versión 2.8.2 de Apache Kafka y es compatible con los clientes Apache Kafka de código abierto.

[La versión 2.8.2 por niveles contiene una funcionalidad de almacenamiento por niveles que es compatible con las API introducidas en Apache Kafka. KIP-405](#) Para obtener más información acerca de la característica por niveles de Amazon MSK, consulte [Almacenamiento por niveles para agentes Standard](#).

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 Apache y de administración. ZooKeeper Amazon MSK proporciona ZooKeeper puntos de enlace TLS, que puede consultar con la operación.

[DescribeCluster](#)

El resultado de la [DescribeCluster](#) operación 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 Amazon con MSK-only corrección de errores de la versión 2.4.1 de Apache Kafka. Esta versión de corrección de errores contiene una solución para [KAFKA-9752](#) un problema poco frecuente que provoca que los grupos de consumidores se reequilibrén 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 [Actualización de 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` en los ID de las zonas de disponibilidad de los agentes. 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 “PER_TOPIC_PER_BROKER Supervisión de niveles”](#).

Para obtener información sobre cómo encontrar los ID de las zonas de disponibilidad, consulte los ID de zona [de disponibilidad de su recurso](#) en la guía del AWS Resource Access Manager usuario.

Para obtener información sobre los ajustes de las propiedades de configuración, consulte [the section called “Configuraciones de agentes”](#).

Para obtener más información al respecto KIP-392, consulta [Permitir a los consumidores recuperar la réplica más cercana](#) 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 [Actualización de 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](#)
- [Actualización de 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 conocer los detalles sobre la actualización de la versión de Apache Kafka, consulte [Actualización de 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 de versión automáticas pueden ocurrir en cualquier momento después de la fecha de fin de soporte. No recibirá ninguna notificación antes de la actualización de versión.

- MSK eliminará gradualmente el soporte para los clústeres recién creados que utilicen versiones de Kafka con fechas de fin de soporte publicadas.

Actualización de la versión de Apache Kafka

Puede actualizar un clúster de Amazon MSK existente a una versión más reciente de Apache Kafka. Antes de actualizar la versión de Kafka del clúster, verifique que la versión del software del lado del cliente admita las características de la nueva versión de Kafka.

Para obtener información sobre cómo mantener un clúster con alta disponibilidad durante una actualización, consulte [the section called “Crear clústeres de alta disponibilidad”](#).

Actualice la versión de Apache Kafka con el Consola de administración de AWS

1. Abra la consola de Amazon MSK en <https://console.aws.amazon.com/msk/>.
2. En la barra de navegación, seleccione la región en la que creó el clúster de MSK.
3. Seleccione el clúster de Amazon MSK que desea actualizar.
4. En la pestaña Propiedades, elija Actualizar en la sección Versión de Apache Kafka.
5. En la sección Versión de Apache Kafka, realice lo siguiente:
 - a. En la lista desplegable Elegir versión de Apache Kafka, seleccione la versión de destino a la que desea realizar la actualización. En este ejemplo, elija **3.9.x**.
 - b. (Opcional) Seleccione Ver compatibilidad de versiones para verificar la compatibilidad entre la versión actual del clúster y las versiones disponibles para la actualización. Luego, seleccione Elegir para continuar.

Note

Amazon MSK admite actualizaciones de versión en el lugar para la mayoría de las versiones de Apache Kafka. Sin embargo, al actualizar de una versión de ZooKeeper-based Kafka a una KRaft-based versión, debe crear un clúster nuevo. A continuación, copie los datos en el clúster nuevo y cambie los clientes para que utilicen dicho clúster.

- c. (Opcional) Seleccione la casilla Actualizar la configuración del clúster para aplicar actualizaciones de configuración compatibles con la nueva versión. Esto habilita las nuevas características y mejoras de la versión.

Puede omitir este paso si necesita mantener las configuraciones personalizadas existentes.

 Note

- Server-side las actualizaciones no actualizan automáticamente las aplicaciones cliente.
- Para mantener la estabilidad del clúster, no se admiten reducciones de versión.

- d. Seleccione Actualizar versión para iniciar el proceso.

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
```

La salida 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 [DescribeCluster](#) operación 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.

Actualización de la versión de Apache Kafka mediante la API

1. Invoque la [GetCompatibleKafkaVersions](#) operación para obtener una lista de las versiones de Apache Kafka a las que puede actualizar el clúster.
2. Ejecute la [UpdateClusterKafkaVersion](#) operación 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.
- Configure el valor mínimo de réplicas sincronizadas (`minISR`) con un valor máximo igual a uno menos que el factor de replicación (RF), es decir, $\text{minISR} = (\text{RF}) - 1$. Esto garantiza que el conjunto de réplicas de la partición pueda tolerar que una réplica quede fuera de línea o esté subreplicada.
- 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 los parches necesarios. 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 de Amazon MSK Replicator](#).

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](#)
- [Under-replicated particiones \(URP\) superiores a 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 mediante el bootstrap de IPv6](#)
- [No se puede acceder al clúster desde dentro AWS: Problemas de red](#)
- [Error en la autenticación: demasiadas conexiones](#)
- [Error de autenticación: la sesión es 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. [Aprovisionamiento del rendimiento de almacenamiento para agentes Standard 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 varios de sus grupos de consumidores están atrapados en un estado de reequilibrio continuo, la causa podría ser un problema con Apache Kafka [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 [Actualización de 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 bloqueado mediante la acción de la API. [RebootBroker](#)

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 obtener un ejemplo de una política que permite esta acción, consulte [Amazon EC2: Permite administrar grupos de seguridad de EC2 asociados con 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](#) una afección que afecte a la versión 2.1.0 de Apache Kafka y provocar un punto muerto en uno o varios intermediarios. 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 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, consulte [the 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.

Under-replicated particiones (URP) superiores a 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 Standard”](#).
- Si `UnderReplicatedPartitions` es consistentemente mayor que 0, incluso durante periodos de poco tráfico, el problema podría ser que ha establecido ACL restrictivas que no conceden acceso al tema a los agentes. 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 cómo configurar ACL, consulte [Autorización y ACL](#) en la 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 las ACL en `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 ACL de red 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 Consola de administración de AWS”](#).

No se puede acceder al clúster mediante el bootstrap de IPv6

Si tiene problemas para conectarse a un clúster mediante las cadenas de arranque de IPv6 proporcionadas, siga estos pasos:

1. Asegúrese de que su cliente tenga asignadas las direcciones IPv4 e IPv6. La aplicación cliente debe ejecutarse en una subred que tenga habilitadas y correctamente configuradas las direcciones IPv4 e IPv6. Compruebe si su VPC tiene un bloque CIDR de IPv4 y un bloque CIDR de IPv6 asociado, confirme que la subred tiene habilitadas las direcciones IPv4 e IPv6 y compruebe que la instancia de EC2 o el entorno de cliente tengan asignadas direcciones IPv4 e IPv6. Para obtener más información, consulte el [direccionamiento IP de sus VPC y subredes](#) en la Guía del usuario de Amazon VPC.
2. Asegúrese de que los puertos IPv6 pertinentes estén presentes en las reglas de entrada y salida del grupo de seguridad. Agregue reglas de entrada para permitir el tráfico en los puertos del clúster desde sus direcciones IPv6 y configure las reglas de salida para permitir el tráfico de IPv6. Para obtener números de puerto específicos, consulte la [información sobre puertos](#) en la documentación de MSK. Recuerde actualizar las reglas de IPv4 e IPv6 si se ejecuta en modo de doble pila. 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.
3. Asegúrese de que la configuración de las propiedades de la JVM sea correcta para la compatibilidad con IPv6. En la aplicación cliente, configúrelo en `true` y `java.net.preferIPv6Addresses` `java.net.preferIPv4Stack` en `false`. Estos parámetros se pueden configurar como propiedades del sistema o como argumentos de JVM. Reinicie la aplicación después de realizar estos cambios para que surtan efecto.

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 “Obtención de los agentes de arranque”](#) 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 “Obtención de los agentes de arranque”](#) *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.

Cliente de Amazon EC2 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 obtener un ejemplo de cómo acceder a un clúster desde una instancia de Amazon EC2 que esté en la misma VPC que el clúster, consulte [the section called “Introducción”](#).

Cliente de Amazon EC2 y clúster de MSK en distintas VPC

Si el equipo cliente y el clúster están en dos VPC diferentes, asegúrese de lo siguiente:

- Las dos VPC están interconectadas.
- El estado de la interconexión está activo.
- Las tablas de enrutamiento de las dos VPC están correctamente configuradas.

Para obtener información acerca de la interconexión de VPC, consulte [Trabajo con interconexiones de VPC](#).

Cliente On-premises

En el caso de un cliente local que esté configurado para conectarse al clúster de MSK mediante Site-to-Site 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-xxxxxxx)`.
- 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 el tráfico). TLS-encrypted

Para obtener más instrucciones Site-to-Site VPN de solución de problemas, consulte [Solución de problemas de Client VPN](#).

Direct Connect

Si el cliente la usa Direct Connect, consulte [Solución de problemas 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](#).

Error de autenticación: la sesión es demasiado corta

El error `Failed authentication ... Session too short` se produce cuando el cliente se intenta conectar a un clúster con credenciales de IAM que están a punto de caducar. Asegúrese de revisar cómo se renuevan las credenciales de IAM. En la mayoría de los casos, las credenciales se reemplazan demasiado cerca del vencimiento de la sesión, lo que provoca problemas en el servidor y errores 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: AmazonMSKFullAccess](#).

No se puede actualizar `KafkaVersionsList` en la configuración de MSK

Al actualizar la [KafkaVersionsList](#) propiedad en el recurso [AWS::MSK::Configuration](#), la actualización falla y muestra 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 CloudFormation pila falla porque la nueva configuración usa el mismo nombre que la configuración existente. Este tipo de actualización requiere un [reemplazo del recurso](#). Para actualizar `KafkaVersionsList` correctamente, también debe actualizar la propiedad [Nombre](#) en la misma operación.

Además, si la configuración está asociada a algún clúster creado con Consola de administración de AWS o AWS CLI, añada lo siguiente al recurso de configuración para evitar [intentos fallidos de eliminación de recursos](#).

UpdateReplacePolicy: Retain

Después de que la actualización se complete 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 de Amazon MSK aprovisionado](#).

Prácticas recomendadas para agentes Standard y Express

Esta sección describe las prácticas recomendadas que se deben seguir para agentes Standard y Express. Para obtener información sobre las prácticas recomendadas del Replicador Amazon MSK, consulte [Prácticas recomendadas](#).

Temas

- [Prácticas recomendadas para agentes Standard](#)
- [Prácticas recomendadas para agentes Express](#)
- [Prácticas recomendadas para clientes de Apache Kafka](#)

Prácticas recomendadas para agentes Standard

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](#).

Consideraciones del cliente

La disponibilidad y el rendimiento de la aplicación no dependen únicamente de la configuración del lado del servidor, sino también de la configuración del lado del cliente.

- Configure los clientes para lograr 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 pueden quedar fuera de servicio tanto por eventos planificados como no planificados, por ejemplo, actualizaciones, aplicación de parches, fallas de hardware y 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 [Prácticas recomendadas para clientes de Apache Kafka](#) para ver los detalles completos.

- 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](#).
- Ejecute pruebas de rendimiento para verificar que las configuraciones del cliente permiten cumplir los objetivos de rendimiento.

Consideraciones del servidor

Dimensione correctamente el clúster: número de particiones por agente Standard

La siguiente tabla muestra el número recomendado de particiones (incluidas las réplicas líder y seguidoras) por agente Standard. El número recomendado de particiones no se aplica de forma obligatoria y constituye una práctica recomendada para escenarios en los que se envía tráfico a todas las particiones de los temas aprovisionados.

Tamaño del bróker	Número recomendado de particiones (incluidas las réplicas de seguidor y líder) por agente	Número máximo de particiones que admiten las operaciones de actualización
<code>kafka.t3.small</code>	300	300
<code>kafka.m5.large</code> o <code>kafka.m5.xlarge</code>	1 000	1500
<code>kafka.m5.2xlarge</code>	2000	3 000
<code>kafka.m5.4xlarge</code> <code>kafka.m5.8xlarge</code> <code>kafka.m5.12xlarge</code> , <code>kafka.m5.16xlarge</code> o <code>kafka.m5.24xlarge</code>	4000	6000
<code>kafka.m7g.large</code> o <code>kafka.m7g.xlarge</code>	1 000	1500

Tamaño del bróker	Número recomendado de particiones (incluidas las réplicas de seguidor y líder) por agente	Número máximo de particiones que admiten las operaciones de actualización
kafka.m7g.2xlarge	2000	3 000
kafka.m7g.4xlarge , kafka.m7g.8xlarge , kafka.m7g.12xlarge o kafka.m7g.16xlarge	4000	6000

Si tiene casos de uso con muchas particiones y bajo rendimiento, en los que el número de particiones es elevado pero no se envía tráfico a todas estas, puede agrupar más particiones por agente, siempre que haya realizado pruebas suficientes (incluidas pruebas de rendimiento) para validar 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 número elevado de particiones también puede hacer que falten métricas de Kafka en CloudWatch y sobre el 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 [the section called “Tipos de agente”](#).

Dimensione correctamente el clúster: número de agentes Standard por clúster

Para determinar el número adecuado de agentes Standard para el clúster de MSK provisionado y comprender los costos, consulte la hoja de cálculo de [dimensionamiento y precios de MSK](#). Esta hoja de cálculo proporciona una estimación para dimensionar un clúster de MSK provisionado y los costos asociados de Amazon MSK, en comparación con un clúster de Apache Kafka

autoadministrado similar basado en 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 por esta hoja son conservadoras y ofrecen un punto de partida para un nuevo clúster de MSK aprovisionado. 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 comprender cómo afecta la infraestructura subyacente al rendimiento de Apache Kafka, consulte [las prácticas recomendadas para ajustar el tamaño de los clústeres de Apache Kafka a fin de optimizar el rendimiento](#) y los costes en el blog sobre macrodatos. AWS 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 conviene escalar verticalmente frente a escalar horizontalmente, así como orientación sobre cómo verificar de forma continua el tamaño de los clústeres de producción. Para obtener información sobre clústeres basados en almacenamiento por niveles, consulte [Prácticas recomendadas para ejecutar cargas de trabajo de producción con almacenamiento por niveles de Amazon MSK](#).

Optimización del rendimiento de los clústeres para instancias m5.4xl, m7g.4xl o más grandes

Al utilizar instancias m5.4xlarge, m7g.4xlarge o superiores, puede optimizar el rendimiento del clúster de MSK aprovisionado si ajusta las configuraciones `num.io.threads` y `num.network.threads`.

`Num.io.threads` es el número de subprocesos que un agente Standard utiliza para procesar 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 Standard 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.

En la siguiente tabla, se describe la configuración recomendada para cada tamaño de instancia.

Tamaño de instancia	Valor recomendado para num.io.threads	Valor recomendado para num.network.threads
m5.4xl	16	8
m5.8xl	32	16
m5.12xl	48	24
m5.16xl	64	32
m5.24xl	96	48
m7g.4xlarge	16	8
m7g.8xlarge	32	16
m7g.12xlarge	48	24
m7g.16xlarge	64	32

Utilice la versión más reciente de Kafka para evitar problemas de discordancia en AdminClient los identificadores de los

El identificador de un tema se pierde (error: no coincide con el identificador del tema de la partición) cuando se utiliza una AdminClient versión de Kafka anterior a la 2.8.0 con el indicador para aumentar o reasignar las particiones de temas `--zookeeper` para un clúster aprovisionado por MSK que utiliza 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

Utilice las siguientes recomendaciones para que los clústeres de MSK aprovisionados mantengan una alta disponibilidad durante una actualización (por ejemplo, al actualizar el tamaño del agente o la versión de Apache Kafka) o cuando Amazon MSK reemplaza 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.

Supervisión del uso de CPU

Amazon MSK recomienda firmemente mantener la utilización de CPU de los agentes (definida como $\text{CPU User} + \text{CPU System}$) por debajo del 60 %. Esto garantiza que el clúster retenga suficiente margen de CPU para manejar eventos operativos, como fallas de agentes, aplicación de parches y actualizaciones graduales.

Apache Kafka puede redistribuir la carga de CPU entre los agentes del clúster cuando resulta necesario. Por ejemplo, cuando Amazon MSK detecta y se recupera de una falla en un agente, realiza tareas de mantenimiento automático, como la aplicación de parches. De forma similar, cuando un usuario solicita un cambio en el tamaño del agente o una actualización de versión, Amazon MSK inicia flujos de trabajo graduales que dejan fuera de servicio un agente a la vez. 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. Al seguir esta práctica recomendada, se garantiza un margen de CPU suficiente para tolerar estos eventos operativos.

Note

Al supervisar la utilización de la CPU, tenga en cuenta que el uso total de CPU incluye más componentes que `CPU User` y `CPU System`. Otras categorías, como `iowait`, `irq`, `softirq` y `steal`, también contribuyen a la actividad total de la CPU. En consecuencia, `CPU inactiva` no siempre es igual a $100\% - \text{CPU User} - \text{CPU System}$.

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 la alarma, 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 el clúster de Amazon MSK aprovisionado. Para ello, agregue 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 usar [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 controle 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 los `HeapMemoryAfterGC` aumentos superen el 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 caso de los clústeres aprovisionados por MSK ZooKeeper basados en ZooKeeper comandos de Apache para añadir agentes, estos agentes no se añadirán a su clúster aprovisionado por MSK y su Apache ZooKeeper contendrá información incorrecta sobre el clúster. Esto podría provocar la pérdida de datos. Para conocer las operaciones compatibles de un clúster de MSK aprovisionado, 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 dentro del mismo clúster de MSK aprovisionado, puede usar la herramienta de reasignación de particiones denominada `kafka-reassign-partitions.sh`. Se recomienda no reasignar más de 10 particiones en una sola llamada `kafka-reassign-partitions` para realizar operaciones seguras. 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 agregar agentes a un clúster de MSK aprovisionado, consulte [the section called “Ampliación de un clúster”](#). Para obtener información sobre cómo eliminar agentes de un clúster de MSK aprovisionado, 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 agentes Express

Este tema describe algunas prácticas recomendadas que se deben seguir cuando se utilizan agentes Express. Los agentes Express vienen preconfigurados para ofrecer alta disponibilidad y durabilidad. De forma predeterminada, los datos se distribuyen en tres zonas de disponibilidad; la replicación siempre se establece en 3 y el mínimo de réplicas en sincronía siempre se establece en 2. Sin embargo, aún hay algunos factores que debe tener en cuenta para optimizar la fiabilidad y el rendimiento del clúster.

Consideraciones del cliente

La disponibilidad y el rendimiento de la aplicación dependen no solo de la configuración del servidor, sino también de la configuración del lado del cliente.

- Configure los clientes para lograr 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 pueden quedar fuera de servicio tanto por eventos planificados como no planificados, por ejemplo, actualizaciones, aplicación de parches, fallas de hardware y 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 los detalles completos en [recomendaciones de prácticas recomendadas para clientes de Apache Kafka](#).
- Ejecute pruebas de rendimiento para verificar que las configuraciones del cliente permitan cumplir los objetivos de rendimiento incluso cuando se reinicien los agentes bajo carga máxima. Puede reiniciar los agentes de su clúster desde la consola de MSK o mediante MSK. APIs

Consideraciones del servidor

Temas

- [Ajuste el tamaño correcto de su clúster: número de agentes por clúster](#)
- [Supervisión del uso de CPU](#)
- [Dimensione correctamente el clúster: número de particiones por agente Express](#)
- [Supervisión del número de conexiones](#)
- [Reasignar particiones](#)

Ajuste el tamaño correcto de su clúster: número de agentes por clúster

Elegir la cantidad de agentes para el clúster basado en Express es sencillo. Cada agente Express incluye una capacidad de rendimiento definida para el ingreso y la salida. Debe utilizar esta capacidad de rendimiento como el principal medio para dimensionar el clúster (y luego considerar otros factores, como la cantidad de particiones y de conexiones, que se describen más adelante).

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 la siguiente tabla para conocer nuestros límites de rendimiento recomendados para cada tamaño de agente Express. Si el rendimiento supera los límites recomendados, puede experimentar una degradación del rendimiento y debería reducir el tráfico o escalar el clúster. Si el rendimiento supera

los límites recomendados y alcanza la cuota por agente, MSK aplicará una limitación controlada al tráfico de cliente para evitar una sobrecarga adicional.

También puede consultar la hoja de cálculo [Dimensionamiento y precios de MSK](#) para evaluar varios escenarios y considerar otros factores, como el número de particiones.

La siguiente tabla enumera el rendimiento máximo recomendado por agente para cada tamaño de instancia.

Tamaño de instancia	MBpsEntrada ()	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

Supervisión del uso de CPU

Recomendamos que mantenga la utilización total de la CPU de los agentes (definida como CPU de usuario + CPU del sistema) 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 no planificados. Un ejemplo de un evento planificado es una actualización de la versión del clúster, durante la cual MSK actualiza los agentes del clúster reiniciándolos uno a uno. Un ejemplo de un evento no planificado es una falla de hardware en un agente o, en el peor de los casos, una falla de una zona de disponibilidad (AZ) en la que se ven afectados todos los agentes de esa AZ. Cuando los agentes con réplicas líderes de partición quedan fuera de línea, Apache Kafka reasigna el liderazgo de las particiones para redistribuir el trabajo a otros agentes del clúster. Al seguir esta práctica recomendada, puede asegurarse de que el clúster tenga suficiente margen de CPU 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 del agente](#) al siguiente tamaño mayor. 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 el clúster mediante la adición de agentes](#), y luego reasigne las particiones existentes mediante la herramienta de reasignación de particiones llamada `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 agentes presentan de forma constante una utilización desigual de la CPU, podría ser un indicio de que la carga no se distribuye de manera uniforme dentro del clúster. Recomendamos usar [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 recopilación de JMX. Si habilita la supervisión abierta con la característica de Prometheus, se recomienda usar un intervalo de recopilación de 60 segundos o superior (`scrape_interval: 60s`) para la configuración del host de Prometheus (`prometheus.yml`). Reducir el intervalo de raspado puede provocar un uso elevado de la CPU en el clúster.

Dimensione correctamente el clúster: número de particiones por agente Express

Si tiene casos de uso con muchas particiones y bajo rendimiento, en los que el número de particiones es alto pero no se envía tráfico a todas estas, puede concentrar más particiones por agente, siempre que haya realizado pruebas suficientes (incluidas pruebas de rendimiento) para validar 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
- Asocia un AWS Secrets Manager secreto a un clúster que tenga SASL/SCRAM autenticación

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](#).

Para obtener información sobre el número recomendado de particiones (incluidas las réplicas líder y seguidoras) para cada agente Express, consulte [Cuota de particiones de agentes Express](#). El número recomendado de particiones no se aplica de forma obligatoria y constituye una práctica recomendada para escenarios en los que se envía tráfico a todas las particiones de los temas aprovisionados.

Supervisión del número de conexiones

Las conexiones de los clientes a los agentes consumen recursos del sistema, como memoria y CPU. Según el mecanismo de autenticación que utilice, debe supervisar para asegurarse de que se mantenga 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 reintente las conexiones después de 1 segundo, establezca `reconnect.backoff.ms` en 1000. Para obtener más información sobre cómo configurar los reintentos, consulte la [documentación de Apache Kafka](#).

Dimensión	Cuota
Máximo de conexiones TCP por agente (control de acceso de IAM)	3 000
Máximo de conexiones TCP por agente (IAM)	100 por segundo
Máximo de conexiones TCP por agente (sin IAM)	MSK no aplica límites de conexión para la autenticación sin IAM. No obstante, debe supervisar otras métricas, como el uso de CPU y memoria, para asegurarse de no sobrecarg

Dimensión	Cuota
	ar el clúster debido a un número excesivo de conexiones.

Reasignar particiones

Para mover particiones a distintos agentes dentro del mismo clúster de MSK aprovisionado, puede usar la herramienta de reasignación de particiones denominada `kafka-reassign-partitions.sh`. Recomendamos no reasignar más de 20 particiones en una sola llamada `kafka-reassign-partitions`, a fin de realizar operaciones seguras. 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 agregar agentes a un clúster de MSK aprovisionado, consulte [the section called “Ampliación de un clúster”](#). Para obtener información sobre cómo eliminar agentes de un clúster de MSK aprovisionado, 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 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 consultar prácticas recomendadas para agentes estándar y Express, consulte [Prácticas recomendadas para agentes 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 configurarlo siempre para lograr comportamientos más deterministas y una mejor correlación de registros durante la resolución de problemas. `client/server`
- 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 se traducen en más subprocesos y operaciones a la vez. I/O 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, como mínimo, igual al número de particiones para lograr un mejor paralelismo y mayor resiliencia. En algunas situaciones, puede optar por tener menos consumidores que particiones en temas con 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 despliegue de la aplicación no se reinicie de una vez, sino preferiblemente producers/consumers 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 es normal durante las implementaciones y la conmutación por error del broker.

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.

Al aumentar los tickets de soporte AWS, incluya cualquier valor anormal observado 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

Los errores transitorios con reintentos no constituyen un motivo de preocupación, ya que forman parte del protocolo de Kafka para gestionar problemas temporales, como conmutaciones por error del líder o retransmisiones de red. `record-send-rate` confirma si los productores continúan el procesamiento mediante reintentos.

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 tasas elevadas de obtención y de confirmación generan 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

Una conexión alta creation/termination 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 for Apache Flink para el procesamiento de Flink-based transmisiones de Apache
- AWS Lambda para el procesamiento de eventos

Note

MSK sin servidor requiere el control de acceso de IAM para todos los clústeres. Las listas de control de acceso (ACL) de Apache Kafka no se admiten. 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](#)
- [Configure el tipo de red de doble pila](#)
- [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. Inicie sesión en y abra la Consola de administración de AWS consola de Amazon MSK en <https://console.aws.amazon.com/msk/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 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 VPC diferentes con clústeres sin servidor de MSK. 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 redes, copie los ID de las subredes y del grupo de seguridad y guárdelos, ya que necesitará esta información más adelante para crear un equipo 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. Seleccione 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:

- *region* con el código del Región de AWS lugar donde creó el clúster.
- Ejemplo de ID de cuenta *123456789012*, 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.

JSON

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

```
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "kafka-cluster:Connect",
      "kafka-cluster:DescribeCluster"
    ],
    "Resource": [
      "arn:aws:kafka:us-east-1:123456789012:cluster/msk-serverless-
tutorial-cluster/c07c74ea-5146-4a03-add1-9baa787a5b14-s3"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "kafka-cluster:CreateTopic",
      "kafka-cluster:WriteData",
      "kafka-cluster:DescribeTopic"
    ],
    "Resource": [
      "arn:aws:kafka:us-east-1:123456789012:topic/msk-serverless-
tutorial-cluster/*"
    ]
  }
]
```

Para obtener instrucciones sobre cómo escribir políticas seguras, consulte [the 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. Elija Crear rol.

3. En Casos de uso comunes, elija EC2, y, 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. Elija 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 instancia de Amazon EC2 para utilizarla como equipo 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. Abra la consola de Amazon EC2 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, seleccione la casilla de la fila que representa la instancia de Amazon EC2 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

Después de extraer el archivo de Kafka, asegúrese de que los scripts del directorio `bin` 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 IAM de Amazon MSK.

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 crear temas](#)
- [Creación de un tema y escritura de datos en este](#)

Configuración del entorno para crear temas

- Antes de crear un tema, asegúrese de haber descargado el archivo JAR de AWS MSK IAM en el directorio de la instalación de Kafka. `libs/` Si aún no ha realizado este paso, ejecute el siguiente comando en el directorio `libs/` de Kafka.

```
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 el clúster de MSK sin servidor.

- 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 entorno `CLASSPATH` 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 mediante el comando completo de Java con `classpath` explícito, 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
```

Creación de un tema y escritura de datos en este

1. En el siguiente `export` comando, *my-endpoint* sustitúyalo por la cadena `bootstrap-server` que guardó 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 experimenta problemas `classpath` al ejecutar estos comandos, asegúrese de ejecutarlos desde el directorio correcto. Además, asegúrese de que el JAR de IAM de AWS MSK esté en el directorio. `libs` Como alternativa, puede ejecutar los comandos de Kafka mediante el comando completo de Java con `classpath` explícito, 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. Abra la consola de Amazon MSK en <https://console.aws.amazon.com/msk/home>.
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. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
2. En la lista de instancias de Amazon EC2, elija el equipo cliente que creó 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	Editabile	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 [Topic-level las configuraciones 3.2](#) en la documentación oficial de Apache Kafka para obtener más información y ejemplos de cómo configurarlas.

Note

No puede modificar la configuración `segment.bytes` para los temas en MSK sin servidor. Sin embargo, una aplicación de Kafka Streams puede intentar crear un tema interno con un valor de configuración `segment.bytes` distinto del que MSK sin servidor permite. Para obtener información sobre cómo configurar Kafka Streams con MSK sin servidor, consulte [Uso de Kafka Streams con agentes Express de MSK y Amazon MSK sin servidor](#).

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.

Configure el tipo de red de doble pila

Amazon MSK admite el tipo de red de doble pila para los clústeres MSK Serverless existentes que utilizan la versión 3.6.0 o posterior de Kafka sin coste adicional. Con las redes de doble pila, sus clústeres pueden utilizar direcciones IPv4 e IPv6. Dual-stack Los puntos finales también admiten IPv4, por lo que mantienen la compatibilidad con versiones anteriores. Amazon MSK ofrece compatibilidad con IPv6 a través del tipo de red de doble pila, no como. IPv6-only

De forma predeterminada, los clientes se conectan a los clústeres de Amazon MSK mediante el tipo de red IPv4. Todos los clústeres nuevos que cree también utilizan IPv4 de forma predeterminada.

Para actualizar el tipo de red de un clúster a uno de doble pila, asegúrese de cumplir los requisitos previos que se describen en la siguiente sección. A continuación, utilice la [UpdateConnectivityAPI](#) para actualizar la conectividad a la doble pila.

 Note

Una vez que actualice el clúster para usar el tipo de red de doble pila, no podrá volver a cambiarlo al tipo de red IPv4.

Temas

- [Requisitos previos para usar un tipo de red de doble pila](#)
- [Permisos de IAM para MSK Serverless](#)
- [Utilice un tipo de red de doble pila para un clúster](#)
- [Consideraciones sobre el uso de un tipo de red de doble pila](#)

Requisitos previos para usar un tipo de red de doble pila

Antes de configurar el tipo de red de doble pila para los clústeres, asegúrese de que todas las subredes que proporcione durante la creación del clúster sean compatibles con el tipo de red de doble pila. Si ni siquiera una subred de tu clúster admite la doble pila, no podrás actualizar el tipo de red del clúster a la de doble pila.

Permisos de IAM para MSK Serverless

Debe tener los siguientes permisos de IAM:

- `ec2:DescribeSubnets`
- `ec2:ModifyVpcEndpoint`

[Para obtener una lista completa de los permisos necesarios para realizar todas las acciones de Amazon MSK, consulte la política AWS administrada: AmazonMSKFullAccess.](#)

Utilice un tipo de red de doble pila para un clúster

Puede actualizar el tipo de red de un clúster sin servidor de MSK mediante el Consola de administración de AWS, el CLI, o el SDK.

Using Consola de administración de AWS

1. [Abra la consola Amazon MSK en? https://console.aws.amazon.com/msk/home region=us-east-1#/home/](https://console.aws.amazon.com/msk/home region=us-east-1#/home/).
2. Elija el clúster MSK Serverless para el que desee configurar el tipo de red de doble pila.
3. En la página de detalles del clúster, elija Propiedades.
4. En Configuración de red, selecciona Editar tipo de red.
5. En Tipo de red, selecciona Doble pila.
6. Seleccione Save changes (Guardar cambios).

Using AWS CLI

Puedes usar la API de [actualización y conectividad para](#) actualizar el tipo de red de tu clúster MSK Serverless existente a un clúster de doble pila. En el siguiente ejemplo, se usa el `update-connectivity` comando para establecer el tipo de red del clúster como de doble pila.

En el siguiente ejemplo, sustituya el ARN del clúster de muestra, `arn:aws:kafka:us-east-1::cluster//`, por el ARN del clúster de MSK `123456789012` actual. `myCluster` `12345678-1234-1234-1234-123456789012-1` [Para obtener la versión actual del clúster, usa el comando describe-cluster](#).

```
aws kafka update-connectivity \
  --cluster-arn "arn:aws:kafka:us-
east-1:123456789012:cluster/myCluster/12345678-1234-1234-123456789012-1" \
  --current-version "KTVPDKIKX0DER" \
  --connectivity-info '{
    "networkType": "DUAL"
  }
```

Using AWS SDK

En el siguiente ejemplo, se usa la [UpdateConnectivity](#) API para establecer el tipo de red del clúster como de doble pila.

En el siguiente ejemplo, sustituya el ARN del clúster de muestra, `arn:aws:kafka:us-east-1::cluster//`, por el ARN del clúster de MSK `123456789012` actual. `myCluster` `12345678-1234-1234-1234-123456789012-1` Para obtener la versión actual del [DescribeCluster](#) clúster, usa la API.


```
import boto3

client = boto3.client("kafka")

response = client.update_connectivity(
    ClusterArn="arn:aws:kafka:us-
east-1:123456789012:cluster/myCluster/12345678-1234-1234-1234-123456789012-1",
    CurrentVersion="KTVDPKIKX0DER",
    ConnectivityInfo={
        "NetworkType": "DUAL "
    }
)

print("Connectivity update initiated:", response)
```

Consideraciones sobre el uso de un tipo de red de doble pila

- Actualmente, la compatibilidad con IPv6 solo está disponible en el modo de doble pila (IPv4 + IPv6), no como. IPv6-only
- Dual-stack el tipo de red no está disponible para la conectividad privada de varias VPC.
- Puede cambiar el tipo de red de IPv4 a doble pila para un clúster existente solo si todas sus subredes admiten el tipo de red de doble pila.
- No puede volver al tipo de red IPv4 después de habilitar la doble pila. Para volver a la versión anterior, debe eliminar y volver a crear el clúster.
- Debe tener los siguientes permisos de IAM:
 - `ec2:DescribeSubnets` y `ec2:ModifyVpcEndpoint`

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: $m1 * 60 / \text{PERIOD}(m1)$

Métricas disponibles en el nivel de supervisión DEFAULT

Name	Cuando está visible	Dimensiones	Description (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.

Name	Cuando está visible	Dimensiones	Description (Descripción)
MessagesInPerSec	Después de que un productor escribe en un tema	Nombre del clúster, tema	El número de mensajes entrantes por segundo para el 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 Consola de administración de AWS 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. Seleccione AWS/Kafka /Nombre del clúster, tema o AWS/Kafka/Nombre del clúster, grupo de consumidores o tema para ver las 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 versiones de Kafka Connect 2.7.1 o 3.7.x, que son marcos de trabajo de código abierto para conectar 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 gestionados 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”](#), y [the 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 usar

la operación de la UpdateConnector API Amazon MSK Connect para escalar verticalmente hacia arriba o hacia abajo la v CPUs entre 1 y 8 v CPUs para admitir un rendimiento variable.

- Conectividad de red privada: Amazon MSK Connect se conecta de forma privada a los sistemas de origen AWS PrivateLink y destino mediante nombres DNS privados.

Introducción a MSK Connect

Este es un step-by-step tutorial que utiliza el Consola de administración de AWS para crear un clúster de MSK y un conector receptor 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 actúa como destino y recibe 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.
- Una política de IAM que contiene los permisos para escribir en el bucket de destino de S3.
- Un rol de IAM que permite al conector escribir en el bucket de S3 de destino. Agregará la política de IAM que cree 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 consola de Amazon S3 Consola de administración de AWS y ábrala 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 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, elija 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 VPC y las subredes de Amazon 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.

Para crear 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 Editor de políticas, elija JSON y, a continuación, reemplace el JSON en la ventana del editor por el siguiente JSON.

En el siguiente ejemplo, *<amzn-s3-demo-bucket-my-tutorial>* sustitúyalo por el nombre del bucket de S3.

JSON

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

Para obtener instrucciones sobre cómo escribir políticas seguras, consulte [the section called “Control de acceso de IAM”](#).

5. Elija Siguiente.

6. En la página Revisar y crear, haga lo siguiente:
 - a. En Nombre de la política, introduzca un nombre descriptivo, como **mkc-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, agregue una etiqueta a la política con el par clave-valor de **Environment** y **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 Roles y, a continuación, seleccione Crear rol.
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 S3.
 - c. En Caso de uso, elija S3.
3. Elija Siguiente.
4. En la página Agregar permisos, haga lo siguiente:
 - a. En el cuadro de búsqueda de Políticas de permisos, introduzca el nombre de la política que creó previamente para este tutorial. Por ejemplo, mkc-tutorial-policy. A continuación, marque 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 [Creación de roles y asociación de 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 del rol, introduzca un nombre descriptivo, como **mkc-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 tu Cuenta de AWS perfil y no se pueden hacer 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 las secciones Paso 1: Seleccionar entidades de confianza o Paso 2: Agregar permisos, elija 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, agregue una etiqueta al rol con el par clave-valor de **ProductManager** y **John**.

Para obtener más información sobre el uso de etiquetas, consulte [Etiquetas para obtener 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.

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. Seleccione 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. Seleccione 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.

Es posible que tarde AWS unos minutos en terminar de crear el 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 instancia de Amazon EC2 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. Abra la consola de Amazon EC2 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. Abra la consola de Amazon EC2 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](https://archive.apache.org/).

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 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`* reemplácelo 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 Consola de administración de AWS.

Creación del conector

1. ¿Iniciar sesión en la Consola de administración de AWS consola Amazon MSK y abrirla desde <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. En la sección de configuración de red del conector, elija una de las siguientes opciones para el tipo de red:
 - IPv4(predeterminado): IPv4 solo para conectividad con destinos
 - Doble pila: para la conectividad a los destinos a través de ambos IPv4 y IPv6 (solo disponible si las subredes tienen IPv4 bloques IPv6 CIDR asociados a ellas)
8. Copie la siguiente configuración y péguela en el campo de configuración del conector.

Asegúrese de reemplazar la región por el código del Región de AWS lugar donde va a crear el conector. Además, sustituya el nombre `<amzn-s3-demo-bucket-my-tutorial>` del bucket de Amazon S3 por el nombre de su bucket en el siguiente ejemplo.

```
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.partitionner.DefaultPartitioner
storage.class=io.confluent.connect.s3.storage.S3Storage
s3.bucket.name=<amzn-s3-demo-bucket-my-tutorial>
topics.dir=tutorial
```

9. En Permisos de acceso, elija `mkc-tutorial-role`.
10. Elija Siguiente. En la página Seguridad, vuelva a seleccionar Siguiente.
11. En la página Registros, seleccione Siguiente.
12. En la página Revisar y crear, revise la configuración del conector y seleccione 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 Sustitúyalo 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
```

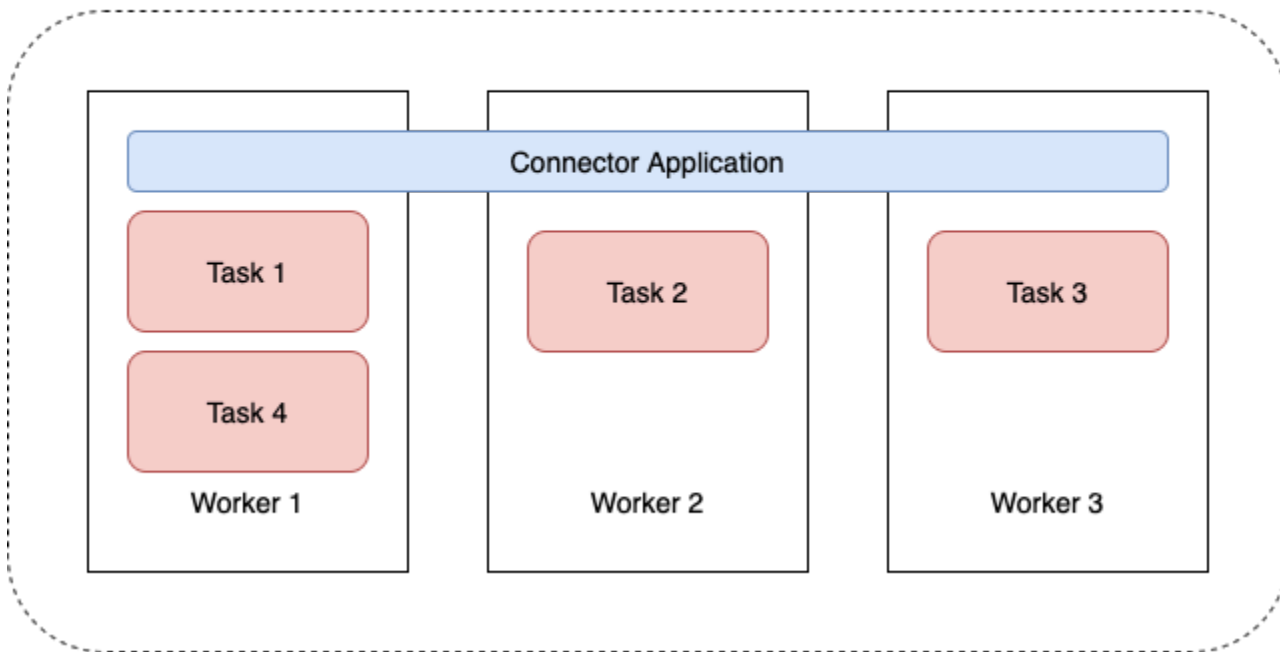
3. Escriba el mensaje que desee y pulse Enter (Entrar). Repita este paso dos o tres veces. Cada vez que introduzca una línea y pulse Enter (Entrar), dicha línea se envía al clúster de Apache Kafka como un mensaje separado.
4. Busque en el bucket de Amazon S3 de los mensajes que envió en el paso anterior.

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.

Connector Architecture



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
- (Opcional) Recuento máximo de tareas de escalado automático: el número máximo de tareas asignadas al conector durante las operaciones de escalado automático. Este parámetro le permite establecer un límite máximo en la creación de tareas, lo que proporciona un mayor control sobre la utilización de los recursos y el paralelismo en relación con las particiones temáticas de Kafka.

Para obtener más información sobre los trabajadores [the section called “Descripción de los procesos de trabajo de MSK Connect”](#), consulte y para obtener más información sobre el número máximo de tareas de escalado automático, consulte [the section called “Conozca el número máximo de tareas de escalado automático”](#) Para obtener más información sobre las métricas de MSK Connect, consulte [the section called “Supervisión de MSK Connect”](#).

Conozca el número máximo de tareas de escalado automático

El `maxAutoscalingTaskCount` parámetro es un campo de capacidad opcional disponible para los conectores de escalado automático en Amazon MSK Connect. Este parámetro le permite establecer un límite máximo en el número máximo de tareas que se pueden crear durante las operaciones de escalado automático de los conectores, lo que proporciona un mayor control sobre la utilización y el rendimiento de los recursos.

Cuando utiliza el modo de capacidad con escalado automático, Amazon MSK Connect anula automáticamente la propiedad `tasks.max` del conector con un valor proporcional al número de

trabajadores MCUs y por trabajador. El `maxAutoscalingTaskCount` parámetro proporciona una opción configurable adicional para limitar el número máximo de tareas creadas para el conector.

Esta capacidad resulta especialmente útil cuando se quiere controlar el nivel de paralelismo en relación con el número de particiones temáticas del clúster de Kafka. Al establecer este límite, puede optimizar el rendimiento y evitar una distribución ineficiente de las tareas que podría producirse cuando el recuento de tareas calculado automáticamente supere los requisitos de carga de trabajo.

Requisitos de configuración

El `maxAutoscalingTaskCount` parámetro debe cumplir los siguientes requisitos:

```
maxAutoscalingTaskCount ≥ maxWorkerCount
```

Este requisito garantiza una utilización eficiente de los recursos al mantener al menos una tarea por trabajador. El sistema aplica este mínimo para optimizar la funcionalidad del conector.

Si lo especifica `maxAutoscalingTaskCount`, el límite se aplica inmediatamente después de la creación del conector y durante todos los eventos de escalado posteriores. A medida que el número de trabajadores aumenta o disminuye durante las operaciones de escalado automático, el sistema sigue respetando este límite. El `tasks.max` valor se ajusta proporcionalmente al número de trabajadores y MCUs por trabajador, pero nunca supera el valor configurado. `maxAutoscalingTaskCount`

Si no especifica este parámetro, el conector utiliza el cálculo estándar sin ningún límite: `tasks.max = workerCount × mcuCount × tasksPerMcu` (donde `tasksPerMcu` es 2).

Cuándo usar `maxAutoscalingTask Count`

Considere la posibilidad de usarlo `maxAutoscalingTaskCount` en los siguientes escenarios:

- Recuento de particiones limitado: si sus temas de Kafka tienen un número fijo de particiones inferior al recuento de tareas calculado automáticamente, establecer un límite impide la creación de tareas inactivas sin trabajo que realizar.
- Optimización del rendimiento: cuando haya identificado que un recuento de tareas específico proporciona un rendimiento óptimo para su carga de trabajo, puede limitar el número máximo de tareas para mantener un rendimiento constante.
- Administración de recursos: cuando desee controlar el paralelismo máximo y el consumo de recursos de su conector, independientemente del número de trabajadores que estén trabajando.

Ejemplo

Para un conector con la siguiente configuración:

```
minWorkerCount: 1
maxWorkerCount: 4
mcuCount: 8
maxAutoscalingTaskCount: 15
```

Sin `maxAutoscalingTaskCount` él, si se escalara a 4 trabajadores, el conector crearía 64 tareas (4 trabajadores × 8 MCUs × 2 tareas por MCU). Si `maxAutoscalingTaskCount` se establece en 15, el conector crea solo 15 tareas, lo que puede ser más adecuado si el tema de Kafka tiene 15 particiones o menos.

Configure el tipo de red de doble pila

Amazon MSK Connect admite el tipo de red de doble pila para los nuevos conectores. Con las redes de doble pila, sus conectores se pueden conectar a destinos a través de una y. IPv4 IPv6 Tenga en cuenta que la IPv6 conectividad solo está disponible en el modo de doble pila (IPv4 + IPv6); no se IPv6 admiten redes exclusivas.

De forma predeterminada, los conectores nuevos utilizan el tipo de IPv4 red. Para crear un conector con un tipo de red de doble pila, asegúrese de cumplir los requisitos previos que se describen en la siguiente sección. Tenga en cuenta que, una vez que haya creado un conector con un tipo de red de doble pila, no podrá modificar su tipo de red. Para cambiar los tipos de red, debe eliminar y volver a crear el conector.

Amazon MSK Connect también admite la conectividad de puntos de conexión de la API de servicio a través de IPv6 y IPv4. Para utilizar la IPv6 conectividad para las llamadas a la API, debe utilizar los puntos de enlace de doble pila. Para obtener más información sobre los puntos de enlace del servicio MSK Connect, consulte los puntos de enlace y las cuotas de [Amazon MSK Connect](#).

Requisitos previos para utilizar un tipo de red de doble pila

Antes de configurar el tipo de red de doble pila para los conectores, asegúrese de que todas las subredes que proporcione durante la creación del conector tengan ambos bloques IPv6 y IPv4 bloques CIDR asignados.

Consideraciones sobre el uso de un tipo de red de doble pila

- IPv6 Actualmente, el soporte solo está disponible en el modo de doble pila (IPv4 + IPv6), no solo como IPv6
- Los conectores con doble pila se pueden conectar tanto IPv4 a los sistemas de datos MSK como IPv6 a los sistemas de datos Sink o Source
- El tipo de red no se puede modificar después de crear el conector; debe eliminarlo y volver a crearlo para cambiar los tipos de red
- Todas las subredes especificadas durante la creación del conector deben admitir la doble pila para que la creación del conector se realice correctamente con un tipo de red de doble pila
- Si se utilizan subredes de doble pila pero no se especifica ningún tipo de red, el conector pasará a ser -only de forma predeterminada por motivos de compatibilidad con versiones anteriores IPv4
- En el caso de los conectores existentes, no puede actualizar el tipo de red; debe eliminar y volver a crear el conector para cambiar los tipos de red
- El uso de redes de doble pila no implica costes adicionales

Creación de un conector de

En este procedimiento, se describe cómo crear un conector desde la Consola de administración de AWS.

Crear un conector mediante el Consola de administración de AWS

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. En la sección de configuración de red del conector, elija una de las siguientes opciones para el tipo de red:
 - IPv4(predeterminado): IPv4 solo para conectividad con destinos
 - Doble pila: para la conectividad a los destinos a través de ambos IPv4 y IPv6 (solo está disponible si las subredes tienen IPv4 bloques IPv6 CIDR asociados a ellas)
8. 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.partitioners.DefaultPartitioner
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.kafka.connect.storage.StringConverter
schema.compatibility=NONE
```

9. 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”](#).
10. (Opcional) En la sección Recuento máximo de tareas de escalado automático, utilice el campo Recuento máximo de tareas de escalado automático para introducir el número máximo de tareas que desea asignar al conector durante las operaciones de escalado automático. El valor debe ser al menos igual al número máximo de trabajadores. Si no especifica ningún valor, el conector utiliza el cálculo estándar sin ningún límite. Para obtener más información, consulte [the section called “Conozca el número máximo de tareas de escalado automático”](#).
11. 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”](#).

12. 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”](#).
13. Seleccione Siguiente, revise la información de seguridad y, a continuación, vuelva a seleccionar Siguiente.
14. Especifique las opciones de registro que desee y elija Enviar. Para obtener más información acerca del registro, consulte [the section called “Registro”](#).
15. En la página Revisar y crear, revise la configuración del conector y seleccione Crear conector.

Para usar la API MSK Connect para crear un conector, consulte [CreateConnector](#).

Puede usar la API `UpdateConnector` para modificar la configuración del conector. Para obtener más información, consulte [the section called “Actualización de un conector”](#).

Actualización de un conector

Este procedimiento describe cómo actualizar la configuración de un conector de MSK Connect existente mediante la Consola de administración de AWS.

Actualización de la configuración del conector mediante el Consola de administración de AWS

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. Elija Editar configuración del conector
5. Actualice la configuración del conector. No puede anular el `connector.class` uso `UpdateConnector` de. El siguiente ejemplo muestra una configuración de ejemplo 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.partitionner.DefaultPartitioner
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.kafka.connect.storage.StringConverter
schema.compatibility=NONE
```

6. Seleccione Enviar.
7. Luego 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. Cuando se crea el complemento, MSK Connect copia el contenido del objeto S3 en ese momento. No mantiene un enlace al objeto S3, por lo que cualquier modificación posterior del objeto no afectará al complemento ni a sus conectores. 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 de uno a varios: puede crear uno o más conectores a partir del mismo complemento.

 Note

Los complementos personalizados no se pueden actualizar in situ. Para usar una nueva versión del código del complemento, elimine todos los conectores que hagan referencia al complemento, elimine el complemento y, a continuación, vuelva a crearlo.

 Paquete de dependencias para complementos personalizados

Te recomendamos que incluyas todos los archivos JAR y las dependencias necesarios para tu complemento. Package el conector como uno de los siguientes:

- Un archivo ZIP que contiene todos los archivos JAR y las dependencias necesarios para el complemento.
- Un único uber JAR que contiene todos los archivos de clases del complemento y sus dependencias.

No agrupar las dependencias del complemento puede afectar a la disponibilidad o la compatibilidad en el entorno de ejecución y provocar errores inesperados.

La siguiente tabla muestra la versión de tiempo de ejecución de Java utilizada para cada versión compatible de Apache Kafka Connect. Asegúrese de que su complemento personalizado sea compatible con la versión de ejecución de Java de la versión de Kafka Connect seleccionada.

Versiones de ejecución de Java para Apache Kafka Connect

Versión Apache Kafka Connect	Tiempos de ejecución de Java
2.7.1	Java 11
3.7.x	Java 17

Para obtener información sobre cómo desarrollar el código de un conector, consulte la [Guía de desarrollo de conectores](#) en la documentación de Apache Kafka.

Creación de un complemento personalizado mediante Consola de administración de AWS

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 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. Debe asegurarse de tener suficientes direcciones IP disponibles en las subredes proporcionadas a una CreateConnector solicitud para tener en cuenta la capacidad especificada, especialmente al escalar automáticamente los conectores, donde la cantidad de trabajadores 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.partition.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
```

Note

Las propiedades de configuración de trabajo se aplican únicamente a las configuraciones de nivel de trabajador de Kafka Connect. Las propiedades de la JVM y del marco de registro (como `log4j.* olog4j2.*`) no son compatibles y no tienen ningún efecto cuando se especifican en una configuración de trabajo personalizada.

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 Consola de administración de AWS.

Crear una configuración de trabajo personalizada mediante Consola de administración de AWS

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 MSK Connect para crear una configuración de trabajo, consulte [CreateWorkerConfiguration](#).

Administre las compensaciones del conector de origen mediante `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.

 Important

Si desea actualizar la configuración de un conector existente y, al mismo tiempo, mantener la continuidad del desfase, utilice la API `UpdateConnector`. Para obtener más información, consulte [Actualización de un conector](#).

Example: Especificar un tema de almacenamiento de compensación al recrear un conector de origen

Si necesita eliminar y volver a crear un conector manteniendo la continuidad de la compensación, puede especificar un tema de almacenamiento de la compensación en su configuración de trabajo. Por ejemplo, supongamos que tiene un conector de captura de datos de cambios (CDC) y quiere volver a crearlo sin perder su lugar en la transmisión de CDC. 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.SecretsManager
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 [Creación de un conector de](#).

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/r0.4.0/msk-config-providers-0.4.0-with-dependencies.zip
```

Extraiga el archivo zip descargado de la carpeta `custom-plugin` con el siguiente comando.

```
unzip msk-config-providers-0.4.0-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.

Amazon S3 > Buckets > msk-lab-...-plugins-bucket > debezium/

debezium/ Copy S3 URI

Objects Properties

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Refresh Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

☐	Name	Type	Last modified	Size	Storage class
☐	custom-plugin.zip	zip	May 15, 2023, 22:43:47 (UTC-04:00)	55.2 MB	Standard

7. Introduzca **debezium-custom-plugin** para el nombre del complemento. De manera opcional, ingrese una descripción y elija Crear complemento personalizado.

Amazon S3 > Buckets > msk-lab-...-plugins-bucket > debezium/

debezium/ Copy S3 URI

Objects Properties

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Refresh Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

☐	Name	Type	Last modified	Size	Storage class
☐	custom-plugin.zip	zip	May 15, 2023, 22:43:47 (UTC-04:00)	55.2 MB	Standard

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

Elija una de las pestañas siguientes para consultar las instrucciones sobre la configuración de 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 usuario.AWS Secrets Manager
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](#). Reemplace el ARN de ejemplo, `arn:aws:secretsmanager:us-east-1:123456789012:secret:MySecret-1234`, por el ARN del 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-1:123456789012: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 [Crear 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

Para configurar objects/files 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](#). Reemplace el ARN de ejemplo, `arn:aws:s3:::<MY_S3_BUCKET/path/to/custom-plugin.zip>`, por el ARN del 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/
truststore_unique_filename.jks}
```

5. Seleccione Usar una configuración personalizada y elija SourceDebeziumCustomConfiguna opción en el menú desplegable Worker Configuration.
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 secciones siguientes, se explica el rol de ejecución del servicio que se debe utilizar con MSK Connect, incluida la política de confianza requerida y los permisos adicionales necesarios al conectarse a un clúster de MSK autenticado con IAM. La página también proporciona ejemplos de políticas integrales de IAM para garantizar el acceso total a la funcionalidad de MSK Connect, así como detalles sobre las políticas AWS gestionadas 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 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 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”](#).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "kafkaconnect.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:kafkaconnect:us-east-1:123456789012:connector/myConnector/abc12345-abcd-4444-a8b9-123456f513ed-2"
        }
      }
    }
  ]
}
```

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 sobre cómo encontrar el UUID del clúster y cómo crear un tema ARNs, consulte. [the section called “Recursos de la política de autorización”](#)

JSON

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

```

        "kafka-cluster:Connect",
        "kafka-cluster:DescribeCluster"
    ],
    "Resource": [
        "arn:aws:kafka:us-east-1:000000000001:cluster/
testClusterName/300d0000-0000-0005-000f-00000000000b-1"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "kafka-cluster:ReadData",
        "kafka-cluster:DescribeTopic"
    ],
    "Resource": [
        "arn:aws:kafka:us-east-1:123456789012:topic/
myCluster/300a0000-0000-0003-000a-00000000000b-6/__amazon_msk_connect_read"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "kafka-cluster:WriteData",
        "kafka-cluster:DescribeTopic"
    ],
    "Resource": [
        "arn:aws:kafka:us-east-1:123456789012:topic/
testCluster/300f0000-0000-0008-000d-00000000000m-7/__amazon_msk_connect_write"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "kafka-cluster:CreateTopic",
        "kafka-cluster:WriteData",
        "kafka-cluster:ReadData",
        "kafka-cluster:DescribeTopic"
    ],
    "Resource": [
        "arn:aws:kafka:us-
east-1:123456789012:topic/testCluster/300f0000-0000-0008-000d-00000000000m-7/
__amazon_msk_connect_*"
    ]
},

```

```

    {
      "Effect": "Allow",
      "Action": [
        "kafka-cluster:AlterGroup",
        "kafka-cluster:DescribeGroup"
      ],
      "Resource": [
        "arn:aws:kafka:us-  
east-1:123456789012:group/testCluster/300d0000-0000-0005-000f-00000000000b-1/  
__amazon_msk_connect_*",
        "arn:aws:kafka:us-  
east-1:123456789012:group/testCluster/300d0000-0000-0005-000f-00000000000b-1/  
connect-*"
      ]
    }
  ]
}

```

Según el tipo de conector, es posible que también necesites adjuntar a la función de ejecución del servicio una política de permisos que le permita acceder a AWS los recursos. 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 la AWS fuente o escribir en el AWS receptor.

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.

JSON

```

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

```

```

    "Action": [
      "kafkaconnect:CreateConnector",
      "kafkaconnect:DeleteConnector",
      "kafkaconnect:DescribeConnector",
      "kafkaconnect:ListConnectors",
      "kafkaconnect:UpdateConnector",
      "kafkaconnect:CreateCustomPlugin",
      "kafkaconnect:DeleteCustomPlugin",
      "kafkaconnect:DescribeCustomPlugin",
      "kafkaconnect:ListCustomPlugins",
      "kafkaconnect:CreateWorkerConfiguration",
      "kafkaconnect:DeleteWorkerConfiguration",
      "kafkaconnect:DescribeWorkerConfiguration",
      "kafkaconnect:ListWorkerConfigurations"
    ],
    "Resource": "*"
  },
  {
    "Sid": "IAMPassRole",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::123456789012:role/MSKConnectServiceRole",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": "kafkaconnect.amazonaws.com"
      }
    }
  },
  {
    "Sid": "EC2NetworkAccess",
    "Effect": "Allow",
    "Action": [
      "ec2:CreateNetworkInterface",
      "ec2:DescribeNetworkInterfaces",
      "ec2:DeleteNetworkInterface",
      "ec2:DescribeVpcs",
      "ec2:DescribeSubnets",
      "ec2:DescribeSecurityGroups"
    ],
    "Resource": "*"
  },
  {
    "Sid": "MSKClusterAccess",
    "Effect": "Allow",

```



```

    "Action": [
      "kafka:DescribeCluster",
      "kafka:DescribeClusterV2",
      "kafka:GetBootstrapBrokers"
    ],
    "Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/myCluster/"
  },
  {
    "Sid": "MSKLogGroupAccess",
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogGroup",
      "logs:CreateLogStream",
      "logs:PutLogEvents",
      "logs:DescribeLogStreams",
      "logs:DescribeLogGroups"
    ],
    "Resource": [
      "arn:aws:logs:us-east-1:123456789012:log-group:/aws/msk-connect/*"
    ]
  },
  {
    "Sid": "S3PluginAccess",
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:ListBucket",
      "s3:PutObject"
    ],
    "Resource": [
      "arn:aws:s3:::amzn-s3-demo-bucket1-custom-plugins",
      "arn:aws:s3:::amzn-s3-demo-bucket1-custom-plugins/*"
    ]
  }
]
}

```

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 puede 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 pueden 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. Sustituya *123456789012* y *arn:aws:kafkaconnect:::connector//* por la información de su conector y. *us-east-1 123456789012 my-S3-Sink-Connector abcd1234-5678-90ab-cdef-1234567890ab* Cuenta de AWS

JSON

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

```
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": " kafkaconnect.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      },
      "ArnLike": {
        "aws:SourceArn": "arn:aws:kafkaconnect:us-
east-1:123456789012:connector/my-S3-Sink-Connector/abcd1234-5678-90ab-
cdef-1234567890ab"
      }
    }
  }
]
```

AWS políticas administradas para MSK Connect

Una política AWS administrada es una política independiente creada y administrada por 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 por 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 asociar la política AmazonMSKConnectReadOnlyAccess a las identidades de IAM.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kafkaconnect:ListConnectors",
        "kafkaconnect:ListCustomPlugins",
        "kafkaconnect:ListWorkerConfigurations"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kafkaconnect:DescribeConnector"
      ],
      "Resource": [
        "arn:aws:kafkaconnect:*:*:connector/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kafkaconnect:DescribeCustomPlugin"
      ],
      "Resource": [
        "arn:aws:kafkaconnect:*:*:custom-plugin/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kafkaconnect:DescribeWorkerConfiguration"
      ]
    }
  ]
}
```

```

        ],
        "Resource": [
            "arn:aws:kafkaconnect:*:*:worker-configuration/*"
        ]
    }
]
}

```

AWS 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 adjuntarse `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.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface"
      ],
      "Resource": "arn:aws:ec2:*:*:network-interface/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/AmazonMSKConnectManaged": "true"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": "AmazonMSKConnectManaged"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [

```

```
    "ec2:CreateNetworkInterface"
  ],
  "Resource": [
    "arn:aws:ec2:*:*:subnet/*",
    "arn:aws:ec2:*:*:security-group/*"
  ],
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:CreateTags"
  ],
  "Resource": "arn:aws:ec2:*:*:network-interface/*",
  "Condition": {
    "StringEquals": {
      "ec2:CreateAction": "CreateNetworkInterface"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeNetworkInterfaces",
    "ec2:CreateNetworkInterfacePermission",
    "ec2:AttachNetworkInterface",
    "ec2:DetachNetworkInterface",
    "ec2>DeleteNetworkInterface"
  ],
  "Resource": "arn:aws:ec2:*:*:network-interface/*",
  "Condition": {
    "StringEquals": {
      "ec2:ResourceTag/AmazonMSKConnectManaged": "true"
    }
  }
}
]
```

MSK Connect actualiza las políticas AWS gestionadas

Consulta los detalles sobre las actualizaciones de las políticas AWS administradas de MSK Connect desde que este servicio comenzó a rastrear estos cambios.

Cambio	Descripción	Fecha
MSK Connect actualizó la política de solo lectura	MSK Connect actualizó la MSKConnect ReadOnlyAccess 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 realizar un seguimiento de los cambios en sus políticas AWS gestionadas.	14 de septiembre de 2021

Uso de roles vinculados a servicios para MSK Connect

Amazon MSK Connect 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 está vinculado directamente a MSK Connect. MSK Connect predefine las funciones vinculadas al servicio e incluyen todos los permisos que el servicio requiere para llamar a AWS otros servicios 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 al servicio en cuestión.

Permisos de roles vinculados a servicios de MSK Connect

MSK Connect utiliza el rol vinculado al servicio denominado —

`AWSServiceRoleForKafkaConnect` Permite que Amazon MSK Connect acceda a los recursos de Amazon en su nombre.

El rol `AWSServiceRoleForKafkaConnect` vinculado al servicio confía en que el servicio asuma el `kafkaconnect.amazonaws.com` 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 Consola de administración de AWS, la o la AWS API AWS CLI, MSK Connect crea automáticamente la función vinculada al servicio.

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 le permite editar el rol vinculado al `AWSServiceRoleForKafkaConnect` 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 AWS CLI o la AWS API 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 Cloud usuario.
- 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 identificarla. 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 instancias de EC2 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 resolución de DNS privado en MSK Connect, los conectores capability/feature utilizaban los solucionadores de DNS de VPC del servicio para las consultas de DNS desde 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 su red](#) en la Guía para desarrolladores de Amazon Route 53 para obtener información sobre el uso de puntos finales 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 registros para su conector, puede ver errores en la creación de conectores debido a consultas de resolución de DNS erróneas en sus CloudWatch 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 has configurado estos CloudWatch registros para tu conector, puedes ver errores en la creación de conectores debido a consultas de resolución de DNS erróneas en tus 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 para MSK Connect

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 MSK-Connect APIs compatible salga de la red de Amazon. Los puntos finales de la interfaz VPC no requieren una puerta de enlace a Internet, un dispositivo NAT, una conexión VPN o una conexión. AWS Direct Connect Para obtener más información, consulte [Uso de las API de Amazon MSK con puntos de conexión 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.

⚠ Important

MSK Connect aplica un límite de velocidad interno a la salida del registro del conector. Si un conector produce un gran volumen de líneas de registro en un período breve, es posible que algunas líneas de registro se eliminen antes de la entrega a cualquier destino configurado. Este límite de velocidad se aplica por igual a CloudWatch los destinos de Logs, Amazon S3 y Amazon Data Firehose. Para evitar la pérdida de registros, diseñe los complementos de sus conectores para minimizar el registro excesivo durante los períodos de ráfaga.

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	Description (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.

El siguiente es 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 Amazon 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 ejecutas AWS 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 usar las siguientes operaciones de la API:

- `DescribeConnectorOperation`: supervise el estado de las operaciones de actualización del conector.
- `ListConnectorOperations`: realice un seguimiento de las actualizaciones anteriores ejecutadas en el 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](#).

Nombre de métrica	Description (Descripción)
<code>CpuUtilization</code>	El porcentaje de consumo de CPU por sistema y usuario.
<code>ErroredTaskCount</code>	El número de tareas que han producido errores.
<code>MemoryUtilization</code>	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 (<code>MemoryUtilization</code>) normalmente comienza con un tamaño de pila mínimo

Nombre de métrica	Description (Descripción)
	que aumenta gradualmente hasta un máximo estable de aproximadamente el 80-90%. El uso en montón de JVM puede aumentar o disminuir 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.
SinkConsumerByteRate	El número promedio de bytes consumidos por segundo por el consumidor de tipo Sink del marco Kafka Connect antes de que se apliquen transformaciones a los datos.
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.

Nombre de métrica	Description (Descripción)
SourceProducerByteRate	El número promedio de bytes producidos por segundo por el productor de tipo Source del marco Kafka Connect después de que se apliquen transformaciones a los datos.
SourceRecordWriteRate	El número medio de salida de registros por segundo desde las transformaciones y escrituras en el clúster de Apache Kafka o Amazon MSK.
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.
BytesInPerSec	Bytes de metadatos transferidos al marco de trabajo Kafka Connect para la comunicación entre nodos de trabajo.
BytesOutPerSec	Bytes de metadatos transferidos desde el marco de trabajo Kafka Connect para la comunicación entre nodos de trabajo.

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.partitionner.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](#)
- [Configuración de los recursos necesarios para MSK Connect](#)
- [Creación del conector](#)
- [Envío de mensajes a Kafka](#)

Requisitos previos

Antes de implementar el conector, asegúrese de contar con 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 permite evitar el tránsito por Internet público sin necesidad de utilizar puertas de enlace NAT.
- Una [máquina cliente](#), como una instancia de Amazon EC2 o [AWS CloudShell](#), para crear temas y enviar registros a Kafka.

Configuración de los recursos necesarios para MSK Connect

Crea un rol de IAM para el conector y, posteriormente, crea el conector. También debe crear una EventBridge regla para filtrar los eventos de Kafka enviados al bus de eventos. EventBridge

Temas

- [Rol de IAM para el conector](#)
- [Una EventBridge regla para los eventos entrantes](#)

Rol de IAM para el conector

La función de IAM que asocie al conector debe tener el [PutEvents](#) permiso para permitir el envío de eventos. EventBridge El siguiente ejemplo de política de IAM le concede 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.

JSON

```
{
  "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 el rol de IAM del conector contenga la siguiente política de confianza.

JSON

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

```

    "Service": "kafkaconnect.amazonaws.com"
  },
  "Action": "sts:AssumeRole"
}
]
}

```

Una EventBridge regla para los eventos entrantes

Crea [reglas](#) que coinciden con los eventos entrantes en función de criterios de datos de eventos, conocidos como [patrones de eventos](#). Con un patrón de eventos, puede definir los criterios para filtrar eventos entrantes y determinar qué eventos deben activar una regla específica y, posteriormente, ser dirigidos 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 objetivo, 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 Consola de administración de AWS.

Temas

- [Paso 1: Descarga del conector](#)
- [Paso 2: creación de un bucket de Amazon S3](#)
- [Paso 3: Creación de un complemento en MSK Connect](#)
- [Paso 4: creación del conector](#)

Paso 1: Descarga del conector

Descargue la versión JAR EventBridge del colector de conectores más reciente de la [página de GitHub versiones](#) del conector EventBridge Kafka. Por ejemplo, para descargar la versión v1.4.1, seleccione el enlace del archivo JAR, `kafka-eventbridge-sink-with-dependencies.jar`, para descargar el conector. A continuación, guarde el archivo en una ubicación de su preferencia en la 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 Consola de administración de AWS, a continuación, elija Amazon S3.
2. En la consola de Amazon S3, seleccione Crear bucket e introduzca un nombre de bucket único. Por ejemplo, **amzn-s3-demo-bucket1-eb-connector**.

3. Seleccione una región adecuada para el bucket de Amazon S3. Asegúrese de que coincida con la región en la que está implementado el clúster de MSK.
4. En Configuración del 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: Creación de un complemento en MSK Connect

1. Abre y Consola de administración de AWS, a continuación, navega hasta MSK Connect.
2. En el panel de navegación izquierdo, seleccione Complementos personalizados.
3. Seleccione Crear complemento y, a continuación, introduzca un Nombre del complemento. Por ejemplo, **eventbridge-sink-plugin**.
4. En Ubicación del complemento personalizado, pegue la URL del objeto de S3.
5. Agregue una descripción opcional para el complemento.
6. Seleccione 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 requerido para evitar errores del conector. Para crear el tema, utilice la máquina cliente.

1. En el panel de navegación izquierdo de la consola de MSK, seleccione Conectores y, a continuación, elija Crear conector.
2. En la lista de complementos, seleccione eventbridge-sink-plugin y, a continuación, seleccione Siguiente.
3. En nombre del conector, introduzca **EventBridgeSink**.
4. En la lista de clústeres, seleccione el clúster de MSK.
5. Copie la siguiente configuración del conector y péguela en el campo Configuración del conector

Reemplace los marcadores de posición en la siguiente configuración según sea necesario.

- Elimine `aws.eventbridge.endpoint.uri` si el 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, modifique la configuración de los procesos de trabajo y del escalado automático. También se recomienda utilizar la versión más reciente disponible (recomendada) de Apache Kafka Connect desde la lista desplegable. En Permisos de acceso, utilice el rol creado anteriormente. También recomendamos activar el registro en CloudWatch para facilitar la observación y solucionar problemas. Ajuste las demás opciones discrecionales, como las etiquetas, según sus necesidades. A continuación, implemente el conector y espere a que el estado pase a En ejecución.

Envío de mensajes a Kafka

Puede configurar codificaciones de mensajes, como Apache Avro y JSON, mediante la especificación de convertidores con el `value.converter` y, de forma opcional, mediante configuraciones de `key.converter` disponibles en Kafka Connect.

El [connector example](#) de este tema está configurado para trabajar con mensajes codificados en JSON, tal como se indica mediante el uso de `org.apache.kafka.connect.json.JsonConverter` para `value.converter`. Cuando el conector se encuentre en estado En ejecución, envíe registros al tema `msk-eventbridge-tutorial` de Kafka desde la 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

Su 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. Aprovisiona 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 identificarla. 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 ayudarlo 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.

Propiedad	Valor
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
- Abra la consola de Secrets Manager en <https://console.aws.amazon.com/secretsmanager/>.
 - 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.
 - Copie el ARN de su secreto.
 - 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.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetResourcePolicy",
        "secretsmanager:GetSecretValue",
        "secretsmanager:DescribeSecret",
        "secretsmanager:ListSecretVersionIds"
      ],
      "Resource": [
        "arn:aws:secretsmanager:us-east-1:123456789012:secret:MySecret-1234"
      ]
    }
  ]
}
```

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 [SecretsManagerConfigProvider](#) la 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"
}

```

Actualización de 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. Ejecute el siguiente AWS CLI comando en la carpeta en la que guardó el archivo JSON en el paso anterior.

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

El siguiente ejemplo muestra la salida cuando el comando se ejecuta 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, en el caso de un conector denominado `file-sink`, el grupo recibe el nombre `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 del conector de origen de Apache Kafka Connect que se ejecutan en las instalaciones o clústeres de Kafka Connect autogestionados que se ejecutan en 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 Kafka 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

del conector Sink de Apache Kafka Connect que se ejecutan en las instalaciones o clústeres de Kafka Connect autogestionados que se ejecutan en 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 Kafka 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 Si ha configurado el `maxAutoscalingTaskCount` parámetro opcional, el `tasks.max` valor no superará este límite. Para obtener más información, consulte [Comprender el número máximo de tareas de ajuste de escala automático](#).
- 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 de Kafka 2.7 ConsumerPartitionAssignor: todas las clases de implementación conocidas](#).

Replicador Amazon MSK

Amazon MSK Replicator es una función totalmente gestionada de Amazon MSK que le permite replicar datos de forma fiable en clústeres de Amazon MSK en clústeres diferentes o iguales. Región de AWS MSK Replicator también admite la replicación de datos desde clústeres autogestionados de Apache Kafka (incluidos los locales, autohospedados u otros proveedores de nube) a clústeres aprovisionados de Amazon MSK con agentes Express. AWS Tanto el clúster de origen como el de destino deben estar en el mismo lugar al replicarse entre clústeres de MSK. Cuenta de AWS Con MSK Replicator, puede crear aplicaciones de streaming resilientes a nivel regional para aumentar la disponibilidad y la continuidad empresarial, y migrar las cargas de trabajo de Apache Kafka desde entornos autogestionados a Amazon MSK. MSK Replicator proporciona una replicación asíncrona automática de datos y compensaciones de grupos de consumidores entre 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. MSK Replicator también replica los metadatos de Kafka necesarios, incluidas las configuraciones de los temas, las listas de control de acceso (ACLs) y las compensaciones por grupos de consumidores. Si se produce un suceso inesperado en una región, puede realizar la conmutación por error a la otra AWS región y reanudar el procesamiento sin problemas.

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 diferentes. AWS En la replicación en la misma región, los clústeres de MSK de origen y de destino se encuentran en la misma región. AWS Debe crear clústeres de MSK de origen y de destino antes de usarlos con el Replicador MSK.

Note

MSK Replicator 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. Oeste (us-west-2, Oregón); Europa (eu-west-1, Irlanda); Europa (eu-central-1, Frankfurt); Asia Pacífico (ap-southeast-1, Singapur); Asia Pacífico (ap-southeast-2, Sídney); Europa (eu-north-1, Estocolmo); Asia Pacífico (ap-south-1, Bombay); Europa (eu-west-3, París); Sudamérica (sa-east-1, São Paulo); Asia Pacífico (ap-northeast-2, Seúl); Europa (eu-west-2, Londres); Asia-Pacífico (ap-northeast-1, Tokio); EE.UU. Oeste (us-west-1, norte de California); Canadá (ca-central-1,

central); China (Beijing) (cn-north-1); China (Ningxia) (cn-northwest-1); y más de 20 regiones adicionales.

Usos comunes de Amazon MSK Replicator:

- 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.
- Distribuya los datos entre sus socios: copie los datos de un clúster de Apache Kafka a varios clústeres de Apache Kafka, de modo que cada uno teams/partners tenga sus propias copias de 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.
- Escriba de forma local y acceda a sus datos en todo el mundo: configure la replicación multiactiva para propagar automáticamente las escrituras realizadas en una AWS región a otras regiones a fin de proporcionar datos con una latencia y un coste más bajos.
- Migre desde clústeres de Kafka autogestionados: migre las cargas de trabajo de Apache Kafka desde implementaciones de Kafka locales, autohospedadas o administradas por terceros a agentes de Amazon MSK Express con sincronización de compensación de grupos de consumidores para una transición perfecta de las aplicaciones.
- Para obtener [the section called “Conceptos clave”](#) más [the section called “Patrones de replicación”](#) información sobre el replicador, consulte y.

Comience a utilizar Amazon MSK Replicator

En esta sección se presenta Amazon MSK Replicator, se describen las configuraciones y los requisitos previos compatibles y se explica cómo configurar su primer replicador.

Temas

- [Configuraciones admitidas](#)
- [Requisitos previos](#)
- [Prepare los clústeres de origen y destino](#)

- [Cree su primer replicador](#)

Configuraciones admitidas

Los siguientes son los requisitos para los tipos de clústeres, las versiones de Kafka, los tipos de instancias y las configuraciones de red cuando se utiliza MSK Replicator.

Tipos y versiones de clústeres compatibles

- MSK Replicator admite clústeres aprovisionados de MSK y clústeres sin servidor de MSK en cualquier combinación como clústeres de origen y de destino.
- MSK Replicator también admite clústeres Apache Kafka autogestionados (Kafka versión 2.8.1 o posterior) con SASL/SCRAM autenticación como clústeres de origen al replicar en clústeres aprovisionados de Amazon MSK con agentes Express. Para obtener más información, consulte [Migre de clústeres de Apache Kafka que no son de MSK a corredores de Amazon MSK Express](#).
- MSK Replicator solo es compatible con los clústeres de MSK que ejecutan Apache Kafka 2.7.0 o superior.
 - La [replicación de nombres de temas idénticos](#) requiere un clúster de MSK que ejecute la versión 2.8.1 o superior de Kafka.
 - Los clústeres autogestionados de Apache Kafka requieren la versión 2.8.1 o posterior de Kafka.
- 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 necesita un mínimo de cuatro agentes en esos clústeres.
- Los clústeres de MSK de origen y de destino deben estar en la misma AWS cuenta. No se admite la replicación en clústeres de cuentas diferentes.

Requisitos de autenticación

- Si utiliza MSK Replicator con un clúster de MSK, el clúster debe tener activado el control de acceso de IAM.
- Si utiliza MSK Replicator con un clúster de Apache Kafka autogestionado como origen, el clúster autogestionado debe tener habilitada la autenticación. SASL/SCRAM

- Para la replicación entre regiones, el clúster de origen debe tener activada la conectividad privada de varias VPC para su método de control de acceso de IAM.
- Para la replicación en la misma región, no se requiere conectividad privada con varias VPC. Otros clientes pueden seguir accediendo al clúster de origen mediante el tipo de autenticación no autenticada.
- Los clústeres sin servidor de MSK requieren el control de acceso de IAM y no admiten la replicación de ACL de Apache Kafka.

Configuración de clústeres sin servidor de MSK

- MSK Serverless admite la replicación de estas configuraciones temáticas para los clústeres de destino durante la creación del tema: `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](#).

Requisitos previos

Antes de crear un replicador de MSK, asegúrese de cumplir los siguientes requisitos previos.

Permisos de IAM obligatorios para crear un Replicador MSK

Antes de crear un replicador de MSK, asegúrese de que su función de IAM tenga los permisos necesarios. Para ver todos los ejemplos de políticas, consulte [Permisos de IAM obligatorios para crear un Replicador MSK](#).

Requisitos previos de red

Según si está configurando la replicación entre regiones o en la misma región, los requisitos de red varían.

- Replicación entre regiones (CRR): el clúster de origen debe tener activada la conectividad privada de varias VPC para el control de acceso de IAM. También debes adjuntar al clúster de origen una política de permisos basada en los recursos. Consulte [Prepare los clústeres de origen y destino](#).
- Replicación en la misma región (SRR): no se requiere conectividad privada con varias VPC. Sin embargo, debe configurar los grupos de seguridad para que el replicador pueda llegar a los clústeres de origen y de destino en el puerto 9098 (el puerto de control de acceso de IAM).

Tanto para el CRR como para el SRR, asegúrese de que la red ACLs no bloquee la conexión entre el MSK Replicator y los clústeres de origen y destino.

Prepare los clústeres de origen y destino

Antes de crear un replicador de MSK, debe preparar un clúster de origen y uno de destino. En esta sección se describen los requisitos para configurar la replicación entre clústeres de Amazon MSK (aprovisionados o sin servidor).

Note

MSK Replicator también admite la replicación entre clústeres autogestionados de Apache Kafka y clústeres aprovisionados de Amazon MSK con agentes Express. Si va a migrar desde una implementación de Kafka autogestionada, consulte [Migre de clústeres de Apache Kafka que no son de MSK a corredores de Amazon MSK Express](#) y para conocer los requisitos previos específicos de los clústeres autogestionados. [Configure los requisitos previos para MSK Replicator con clústeres de Apache Kafka autogestionados](#)

Prepare el clúster de origen

Si ya tiene un clúster de origen de MSK, asegúrese de que cumpla los requisitos descritos en esta sección. De lo contrario, siga estos pasos para crear un clúster de origen aprovisionado o sin servidor de MSK.

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 de 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 o SASL/SCRAM). 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 MSK Serverless, no es necesario que active 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. Puede hacerlo mediante los procedimientos de CLI o de AWS consola que se indican 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.

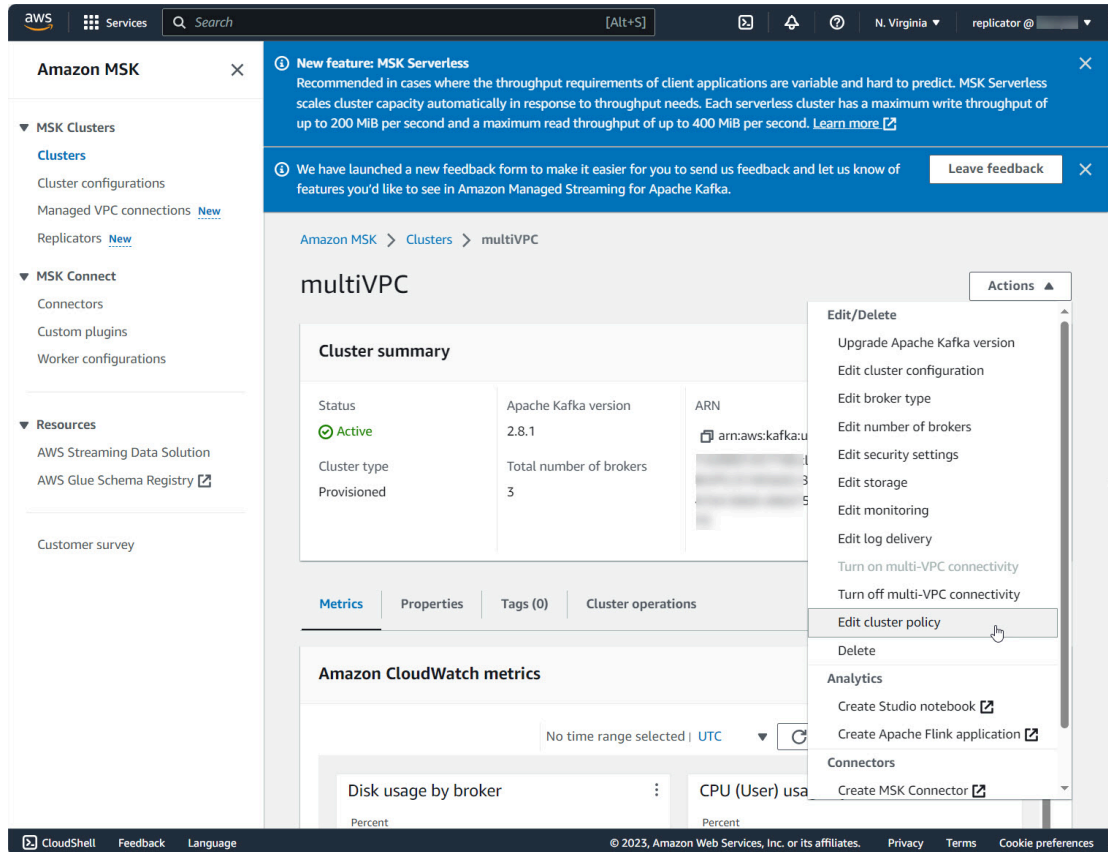
Console: create resource policy

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": "arn:aws:kafka:us-  
east-1:123456789012:cluster/myCluster/abcd1234-5678-90ab-cdef-1234567890ab-1"
    }
  ]
}
```

```
]
}
```

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 para crear un clúster de origen y elige la opción de crear una nueva función de IAM, AWS adjunta la política de confianza necesaria a la función. 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. 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>"  
    }  
  ]  
}'
```

Prepare el clúster de destino

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 conectividad privada de varias VPC. El clúster de destino puede estar en la misma AWS región o en una región diferente que el clúster de origen. Tanto el clúster de origen como el de destino deben estar en la misma AWS cuenta. El clúster de destino debe tener un mínimo de tres agentes.

Cree su primer replicador

En este tutorial, se explica cómo crear su primer replicador de MSK entre dos clústeres de MSK mediante la consola. AWS Para obtener una referencia detallada sobre todas las opciones de creación, consulte. [Cree un replicador mediante la consola AWS](#)

Antes de empezar, asegúrese de que tiene lo siguiente:

- Un clúster MSK de origen con el control de acceso de IAM activado (consulte [Prepare el clúster de origen](#)).
 - Un clúster MSK de destino con el control de acceso de IAM activado (consulte). [Prepare el clúster de destino](#)
 - Los permisos de IAM necesarios (consulte). [Permisos de IAM obligatorios para crear un Replicador MSK](#)
1. [En la AWS región 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 y, a continuación, elija Crear replicador.
 3. En el panel Detalles del replicador, asigne un nombre único al nuevo replicador.
 4. En el panel Clúster de origen, elija la AWS región en la que se encuentra el clúster de origen. Seleccione el clúster de MSK como tipo de clúster y, a continuación, introduzca el ARN del clúster de origen o elija Examinar para seleccionarlo. Las subredes y los grupos de seguridad se rellenarán automáticamente en función de la selección del clúster. Si no se rellenan o si desea utilizar otros, puede seleccionarlos manualmente. Debe seleccionar al menos dos subredes.
 5. En el panel Clúster de destino, elija la AWS región en la que se encuentra el clúster de destino. Seleccione el clúster de MSK como tipo de clúster y, a continuación, introduzca el ARN del clúster de destino o elija Examinar para seleccionarlo. Las subredes y los grupos de seguridad se rellenarán automáticamente en función de la selección del clúster. Si no se rellenan o si desea utilizar otros, puede seleccionarlos manualmente. Debe seleccionar al menos dos subredes.
 6. En el panel de configuración de Replicador, mantenga los valores predeterminados para replicar todos los temas. Elija una configuración de nombre de tema (agregue un prefijo al nombre del tema o mantenga el mismo nombre del tema). Para obtener más información sobre la denominación de los temas, consulte [Denominación de temas \(prefijo o idéntica\)](#).
 7. En el panel de permisos de acceso, seleccione Crear o actualizar la función de IAM con las políticas necesarias para que la consola de MSK cree automáticamente la función de ejecución del servicio.
 8. Seleccione Crear.

El MSK Replicator tarda aproximadamente 30 minutos en crearse correctamente y pasar al estado de ejecución. Puede supervisar el estado en la página Replicadores de la consola de MSK.

Si su MSK Replicator pasa al estado FALLIDO, consulte. [Solución de problemas de Amazon MSK Replicator](#)

Operaciones de replicador

En los temas de esta sección se describe cómo realizar las operaciones habituales del replicador de MSK. Para obtener una lista de todas las operaciones que puede realizar en un replicador de MSK, consulte la referencia de la API de [Amazon MSK](#).

Temas

- [Cree un replicador](#)
- [Edite un replicador](#)
- [Eliminar un replicador](#)
- [Supervisar la replicación](#)

Cree un replicador

Antes de crear un replicador de MSK, asegúrese de tener el [Permisos de IAM obligatorios para crear un Replicador MSK](#) y de haberlo completado. [Prepare los clústeres de origen y destino](#)

Temas

- [Cree un replicador mediante la consola AWS](#)
- [Crear un replicador mediante la CLI AWS](#)
- [Cree un replicador con CloudFormation Terraform](#)

Cree un replicador mediante la consola AWS

Note

Los siguientes pasos se centran en crear un replicador entre dos clústeres de MSK. MSK Replicator también admite la replicación entre clústeres autogestionados de Apache Kafka y clústeres aprovisionados de Amazon MSK con agentes Express. Si va a migrar desde una implementación de Kafka autogestionada, consulte [Migre de clústeres de Apache Kafka que no son de MSK a corredores de Amazon MSK Express](#) y para conocer los requisitos previos

específicos de los clústeres autogestionados. [Configure los requisitos previos para MSK Replicator con clústeres de Apache Kafka autogestionados](#)

Detalles del Replicador

1. [En la AWS región 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 en la que se encuentra el AWS 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 siguiente ejemplo, 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. Seleccione el clúster de MSK como tipo de clúster y, a continuación, introduzca el ARN del clúster de origen o elija Examinar para seleccionarlo.
3. Elija las subredes para el clúster de origen. Las subredes se rellenarán automáticamente en función de la selección del clúster. Si no se rellenan o si desea utilizar otras distintas, puede seleccionarlas manualmente. Debe seleccionar un mínimo de dos subredes. En el caso de un replicador de 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 estar en la misma zona de disponibilidad.
4. Elija los grupos de seguridad para que el Replicador MSK acceda al clúster de origen. Los grupos de seguridad se rellenarán automáticamente en función del clúster que haya seleccionado. Si no se rellenan o si desea utilizar otros, puede seleccionarlos manualmente.

- 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 consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/> y asegúrese de que los grupos de seguridad que proporcionará al replicador tengan reglas de salida que permitan el tráfico a 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.

Para añadir reglas de entrada al grupo de seguridad del clúster de origen:

1. En la AWS consola, ve a los detalles del clúster de origen seleccionando el nombre del clúster.
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.

Para añadir reglas de salida al grupo de seguridad de Replicador proporcionado para la fuente:

1. En la AWS consola de Amazon EC2, vaya al grupo de seguridad que proporcionará durante la creación del replicador para la fuente.
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 utilizar todo el tráfico con origen. `0.0.0.0/0`

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

De forma predeterminada, MSK Replicator crea nuevos temas en el clúster de destino con un prefijo generado automáticamente que se agrega al nombre del tema (por ejemplo,). `<sourceKafkaClusterAlias>.topic` Esto distingue los temas replicados de otros temas del clúster de destino y evita la replicación circular. Puede encontrar el prefijo en el campo `sourceKafkaClusterAlias` mediante la `DescribeReplicator` API o la página de detalles del replicador en la consola de MSK. Como alternativa, puede utilizar la replicación idéntica de los nombres de los temas. Consulte [Denominación de temas \(prefijo o idéntica\)](#).

1. En el panel Clúster de destino, elija la AWS región en la que se encuentra el clúster de destino.
2. Seleccione el clúster de MSK como tipo de clúster y, a continuación, introduzca el ARN del clúster de destino o elija Examinar para seleccionarlo.
3. Elija las subredes para el clúster de destino. Las subredes se rellenarán automáticamente en función de la selección del clúster. Si no se rellenan o si desea utilizar otras distintas, puede seleccionarlas manualmente. Seleccione un mínimo de dos subredes.
4. Elija los grupos de seguridad para que el Replicador MSK acceda al clúster de destino. Los grupos de seguridad se rellenarán automáticamente en función del clúster que haya seleccionado. Si no se rellenan o si desea utilizar otros, puede seleccionarlos manualmente. Para obtener más información sobre el uso de grupos de seguridad, consulte [Controlar el tráfico de sus AWS recursos mediante grupos de seguridad](#) en la Guía del usuario de Amazon VPC.

Tanto para CRR como para SRR, vaya a la consola de Amazon EC2 y asegúrese de que los grupos de seguridad que proporcionará al replicador tengan reglas de salida que permitan el tráfico a 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.

Para añadir reglas de entrada al grupo de seguridad del clúster de destino:

1. En la AWS consola, ve a los detalles del clúster de destino seleccionando el nombre del clúster.
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 de Replicator para el clúster de destino y, a continuación, seleccione Guardar reglas.

Para agregar reglas de salida al grupo de seguridad de Replicator proporcionado para el destino:

1. En la AWS consola, vaya al grupo de seguridad que proporcionará durante la creación del Replicator para el objetivo.
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.
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 utilizar todo el tráfico con el origen. `0.0.0.0/0`

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, cree un replicador independiente. Vaya al Support Center de la AWS consola y [cree un caso de soporte](#) si necesita soporte para más de 750 temas por replicador.

2. De forma predeterminada, MSK Replicator inicia la replicación desde la última compensación (la más reciente). Si desea replicar los datos existentes, también puede iniciar la replicación desde el punto más temprano (el más antiguo). 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.
 - Replicación de nombres de temas idénticos (mantenga el mismo nombre de tema en la consola): los temas se replican con nombres idénticos en el clúster de destino.

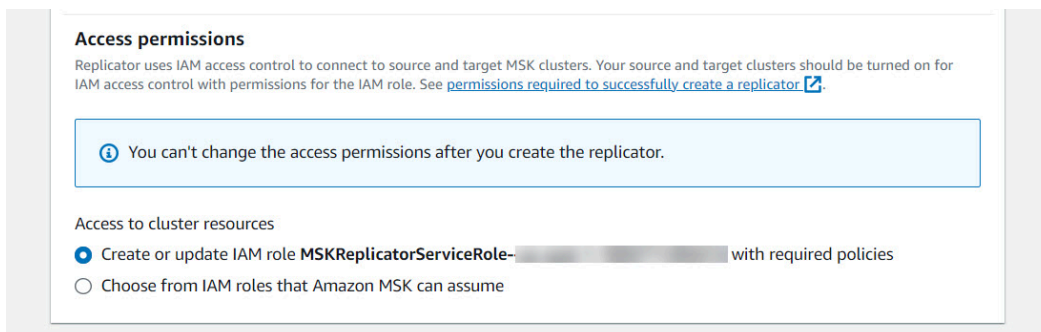
Para obtener más información, consulte [Denominación de temas \(prefijo o idéntica\)](#).

4. De forma predeterminada, MSK Replicator copia todos los metadatos, incluidos los temas, las configuraciones y las compensaciones de los grupos de consumidores ACLs, para lograr una conmutación por error sin problemas. Si no va a crear el replicador para la conmutación por error, si lo desea, puede desactivar una o más de estas configuraciones en la sección Configuración adicional.

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.

También puede configurar el modo de sincronización de compensación del grupo Consumer:

- Legado (predeterminado): las compensaciones se sincronizan cuando los productores escriben en el clúster de origen (unidireccionales).
 - Mejorado: las compensaciones de consumo se sincronizan independientemente de la ubicación del productor (bidireccionales). Requiere un replicador correspondiente que replique los datos del clúster de destino al clúster de origen. Utilice este modo al configurar la replicación bidireccional para migraciones o arquitecturas activo-activas. Para obtener más información, consulte [Sincronización compensada por 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 utiliza la compresión, le recomendamos que utilice el mismo método de compresión que los datos del clúster de origen.
 7. En el panel de permisos de acceso, realice una de las siguientes acciones:
 - a. Seleccione Crear o actualizar el rol de IAM con las políticas necesarias. La consola de MSK adjuntará automáticamente los permisos y la política de confianza necesarios a la función de ejecución del servicio.



- b. Proporcione su propio rol de IAM al seleccionar Elegir de los roles de IAM que puede asumir Amazon MSK. Le recomendamos adjuntar la política de IAM [AWSMSKReplicatorExecutionRole](#) gestionada a su función de ejecución de servicios. Consulte [Función de ejecución de servicios \(SER\)](#).
8. En el panel de entrega de registros, si lo desea, puede configurar la entrega de registros para capturar y enrutar los registros de replicación a los destinos que elija. De forma predeterminada, la entrega de registros no está habilitada. Puede habilitar uno o más de los siguientes destinos:

- Entregue a Amazon CloudWatch Logs: analice, consulte y configure alarmas en los registros.
- Entregue a Amazon S3: almacene y recupere registros sin procesar en el almacenamiento de objetos.
- Entregue a Amazon Data Firehose: capture, transforme y entregue registros a Amazon OpenSearch Service u otros destinos de Amazon Data Firehose.

Para obtener más información, consulte [Registros de MSK Replicator](#).

9. En el panel Etiquetas del Replicador, si así lo quiere, puede asignar etiquetas al recurso del Replicador MSK. En el caso de un Replicador MSK entre regiones, las etiquetas se sincronizan automáticamente con la región remota cuando se crea el replicador.
10. Seleccione Crear.

El MSK Replicator tarda aproximadamente 30 minutos en crearse correctamente y pasar al estado en ejecución. Si su MSK Replicator pasa al estado FALLIDO, consulte. [Solución de problemas de Amazon MSK Replicator](#)

Crear un replicador mediante la CLI AWS

Utilice el `create-replicator` comando para crear un replicador de MSK. Antes de empezar, asegúrese de tener el [Permisos de IAM obligatorios para crear un Replicador MSK](#) y de haberlo completado. [Prepare los clústeres de origen y destino](#)

Note

Los ejemplos siguientes muestran la replicación entre dos clústeres de MSK. MSK Replicator también admite la replicación entre clústeres autogestionados de Apache Kafka y clústeres aprovisionados de MSK con agentes Express. Para ver ejemplos de API que cubren ese escenario, consulte. [CreateReplicator Ejemplos de API para clústeres de Kafka autogestionados](#)

```
aws kafka create-replicator \  
  --replicator-name "<replicator-name>" \  
  --service-execution-role-arn "arn:aws:iam::<account-id>:role/<role-name>" \  
  --kafka-clusters '[  
    {
```

```

    "AmazonMskCluster": {"MskClusterArn": "<source-cluster-arn>"},
    "VpcConfig": {
      "SubnetIds": ["<subnet-1>", "<subnet-2>"],
      "SecurityGroupIds": ["<security-group-id>"]
    }
  },
  {
    "AmazonMskCluster": {"MskClusterArn": "<target-cluster-arn>"},
    "VpcConfig": {
      "SubnetIds": ["<subnet-1>", "<subnet-2>"],
      "SecurityGroupIds": ["<security-group-id>"]
    }
  }
]' \
--replication-info-list '[
{
  "SourceKafkaClusterArn": "<source-cluster-arn>",
  "TargetKafkaClusterArn": "<target-cluster-arn>",
  "TopicReplication": {
    "TopicsToReplicate": [".*"],
    "CopyTopicConfigurations": true,
    "CopyAccessControlListsForTopics": true,
    "DetectAndCopyNewTopics": true,
    "StartingPosition": {"Type": "LATEST"},
    "TopicNameConfiguration": {"Type": "PREFIXED"}
  },
  "ConsumerGroupReplication": {
    "ConsumerGroupsToReplicate": [".*"],
    "ConsumerGroupOffsetSyncMode": "LEGACY"
  },
  "TargetCompressionType": "NONE"
}
]'

```

Para habilitar la entrega de registros al crear un replicador, incluya el `--log-delivery` parámetro. El siguiente ejemplo permite la entrega de CloudWatch registros a Amazon Logs, Amazon S3 y Amazon Data Firehose.

```

aws kafka create-replicator \
  --replicator-name "<replicator-name>" \
  --service-execution-role-arn "arn:aws:iam::<account-id>:role/<role-name>" \
  --kafka-clusters '[
{

```

```

    "AmazonMskCluster": {"MskClusterArn": "<source-cluster-arn>"},
    "VpcConfig": {
      "SubnetIds": ["<subnet-1>", "<subnet-2>"],
      "SecurityGroupIds": ["<security-group-id>"]
    }
  },
  {
    "AmazonMskCluster": {"MskClusterArn": "<target-cluster-arn>"},
    "VpcConfig": {
      "SubnetIds": ["<subnet-1>", "<subnet-2>"],
      "SecurityGroupIds": ["<security-group-id>"]
    }
  }
]' \
--replication-info-list '[
{
  "SourceKafkaClusterArn": "<source-cluster-arn>",
  "TargetKafkaClusterArn": "<target-cluster-arn>",
  "TopicReplication": {
    "TopicsToReplicate": [".*"],
    "CopyTopicConfigurations": true,
    "CopyAccessControlListsForTopics": true,
    "DetectAndCopyNewTopics": true,
    "StartingPosition": {"Type": "LATEST"},
    "TopicNameConfiguration": {"Type": "PREFIXED"}
  },
  "ConsumerGroupReplication": {
    "ConsumerGroupsToReplicate": [".*"]
  },
  "TargetCompressionType": "NONE"
}
]' \
--log-delivery '{
  "ReplicatorLogDelivery": {
    "CloudWatchLogs": {
      "Enabled": true,
      "LogGroup": "/mskr/logs/<log-group-name>"
    },
    "S3": {
      "Enabled": true,
      "Bucket": "<s3-bucket-name>",
      "Prefix": "<optional-prefix>"
    },
    "Firehose": {

```

```
    "Enabled": true,  
    "DeliveryStream": "<delivery-stream-name>"  
  }  
}  
'
```

Puede habilitar uno o más destinos de entrega de registros. Para habilitar solo Amazon CloudWatch Logs, omita los Firehose campos S3 y o establezca su `Enabled` valor en. `false` Para obtener más información sobre la entrega de registros, consulte [Registros de MSK Replicator](#).

Note

Cuando la entrega de registros está habilitada, su función de IAM debe tener los permisos adicionales necesarios para escribir en el destino del registro configurado. Para conocer los permisos necesarios, consulte [Habilitar el registro desde AWS los servicios](#).

Para obtener la referencia completa de la API, consulte [CreateReplicator](#) la referencia de la API de Amazon MSK.

Replicación de temas selectivos

Utilice patrones de expresiones regulares en `topicsToReplicate` y `topicsToExclude` para controlar qué temas se replican. En el siguiente ejemplo, se replican los temas que comienzan por `prod-` y se excluyen los que comienzan por: `test-`

```
"topicReplication": {  
  "topicsToReplicate": ["prod-.*"],  
  "topicsToExclude": ["test-.*"],  
  "detectAndCopyNewTopics": true  
}
```

Verifique el estado del replicador

Después de crear un replicador, compruebe su estado mediante el `describe-replicator` comando:

```
aws kafka describe-replicator \  
  --replicator-arn arn:aws:kafka:us-east-1:123456789012:replicator/my-replicator/xxx
```


El Replicador progresa a través de CREATING → RUNNING estados. Espere aproximadamente 30 minutos hasta que el Replicador alcance RUNNING el estado. Si pasa a FAILED, consulte [Solución de problemas de Amazon MSK Replicator](#).

Cree un replicador con CloudFormation Terraform

AWS CloudFormation

Usa el tipo de `AWS::MSK::Replicator` recurso. Para obtener la referencia completa del recurso, consulte [AWS::MSK::Replicator](#) la Guía del AWS CloudFormation usuario.

Terraform

Utilice el `aws_msk_replicator` recurso. Para ver la referencia completa del recurso, consulte [aws_msk_replicator](#) en el registro de Terraform.

Edite un replicador

No puede cambiar la configuración de replicación del clúster de origen, el clúster de destino, la posición inicial del replicador ni el nombre del tema una vez creado el replicador de MSK. Debe crear un replicador nuevo para cambiar esta configuración. Sin embargo, puede editar otros ajustes del Replicador, como los temas y los grupos de consumidores que desee replicar.

1. ¿Iniciar sesión en la Consola de administración de AWS consola Amazon MSK y abrirla desde <https://console.aws.amazon.com/msk/casa?region=us-east-1#/home/>.
2. En el panel de navegación izquierdo, elija Replicadores y seleccione el replicador de MSK que desee editar.
3. Elija la pestaña Propiedades.
4. En la sección Configuración del replicador, elija Editar replicador.
5. Puede editar la siguiente configuración:
 - Temas para replicar mediante expresiones regulares en las listas de permitidos y rechazados.
 - Configuración de replicación de metadatos (configuraciones de temas ACLs, desfases de grupos de consumidores).

 Note

MSK Replicator no replica la escritura, ACLs ya que sus 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.

- Los consumidores se agrupan para replicar utilizando expresiones regulares en las listas de permitidos y rechazados. Si las listas de permitidos y rechazados están vacías, la replicación de grupos de consumidores está desactivada.
- Configuraciones de entrega de registros (logDelivery) para reenviar los registros operativos a Amazon CloudWatch Logs, Amazon S3 o Amazon Data Firehose. Consulte [Registros de MSK Replicator](#).

 Note

Las configuraciones de entrega de registros no se pueden actualizar al mismo tiempo que la configuración de grupos de consumidores o temas. Estos ajustes se deben actualizar de forma independiente en acciones de actualización independientes.

6. Guarde los cambios. Los cambios tardan aproximadamente 30 minutos en surtir efecto.

Eliminar un replicador

Es posible que tenga que eliminar un replicador de MSK si no se puede crear (estado FALLIDO). Los clústeres de origen y destino asignados a un replicador de MSK no se pueden cambiar una vez creado el replicador de MSK. 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 en la que se encuentra el clúster de destino, ¿debe iniciar sesión y abrir la consola de Amazon MSK en <https://console.aws.amazon.com/msk/casa? Consola de administración de AWS 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 usar [Amazon CloudWatch](#) en la región del clúster de destino para `ReplicationLatency` ver las métricas de cada MSK Replicator y `ReplicatorThroughput` a nivel de tema y agregado. `MessageLag` Las métricas están visibles `ReplicatorName` en el espacio de nombres «AWS/Kafka». También puede ver `ReplicatorFailure` y las `ThrottleTime` métricas para comprobar si hay problemas. `AuthError`

La consola de MSK muestra un subconjunto de CloudWatch métricas para cada replicador de MSK. En la lista Replicadores de la consola, seleccione el nombre de un replicador y, luego, seleccione la pestaña Supervisión.

También puede habilitar el reenvío de registros para su MSK Replicator para obtener una mayor visibilidad de las operaciones de replicación y simplificar la solución de problemas. MSK Replicator admite el reenvío de CloudWatch registros operativos a Amazon Logs, Amazon S3 y Amazon Data Firehose. De forma predeterminada, la entrega de registros no está habilitada. Puede configurar la entrega de registros al crear o actualizar un replicador mediante la `logDelivery` configuración de la solicitud `CreateReplicator` o de la `UpdateReplicator` API.

Para obtener información más detallada sobre la supervisión, las métricas, los registros y la solución de problemas, consulte [Observabilidad](#).

Patrones de replicación

MSK Replicator admite varios patrones de replicación según el caso de uso. En esta sección se describe cómo configurar cada patrón y cómo realizar las operaciones de conmutación por error y conmutación por recuperación.

Temas

- [Replicación en la misma región](#)
- [Replicación entre regiones](#)
- [Replicación activa-pasiva](#)
- [Replicación activa-activa](#)
- [Conmutación por error a la región secundaria AWS](#)
- [Conmutación por error planificada](#)
- [Conmutación por error no planificada](#)

- [Recaída por error](#)

Replicación en la misma región

En la replicación en la misma región (SRR), los clústeres de MSK de origen y de destino se encuentran en la misma región. AWS La replicación en la misma región es útil para agregar datos, distribuirlos a los socios o migrar entre clústeres.

Diferencias clave con respecto a la replicación entre regiones:

- El clúster de origen no requiere conectividad privada de varias VPC.
- No es necesario adjuntar al clúster de origen una política de permisos basada en recursos.
- Debe proporcionar grupos de seguridad para los clústeres de origen y de destino. Las subredes que seleccione para los clústeres de origen y destino deben estar en las mismas zonas de disponibilidad.
- Otros clientes pueden seguir accediendo al clúster de origen mediante el tipo de autenticación no autenticada.

Replicación entre regiones

En la replicación entre regiones (CRR), los clústeres de MSK de origen y destino se encuentran en regiones diferentes. AWS La replicación entre regiones es la base para crear aplicaciones de streaming resilientes para múltiples regiones.

Requisitos clave para la replicación entre regiones:

- El clúster de origen debe tener activada la conectividad privada de varias VPC para el control de acceso de IAM. Consulte [Cluster owner turns on multi-VPC](#).
- Debe adjuntar una política de permisos basada en recursos al clúster de origen. Consulte [Prepare el clúster de origen](#).
- No necesita proporcionar grupos de seguridad para el clúster de origen.
- El replicador se crea en la AWS región del clúster de destino.

La latencia de replicación varía en función de la distancia de red entre las AWS regiones, la capacidad de rendimiento de los clústeres y el número de particiones. Por ejemplo, la latencia de

replicación suele ser menor entre Europa (Irlanda) y Europa (Londres) en comparación con Europa (Irlanda) y Asia Pacífico (Sídney).

Replicación activa-pasiva

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. Las configuraciones activa-pasiva son adecuadas para aplicaciones que solo pueden ejecutarse en una AWS región a la vez o cuando se necesita un mayor control sobre el orden de procesamiento de los datos.

Para una configuración activa-pasiva, recomendamos lo siguiente:

- Opere con una estructura similar de productores, agrupaciones de MSK y consumidores (con el mismo nombre de grupo de consumidores) en dos regiones diferentes. AWS
- Asegúrese de que los dos clústeres de MSK tengan la misma capacidad de lectura y escritura para una replicación de datos fiable.
- Cree un replicador MSK para copiar continuamente los datos del clúster principal al clúster en espera.
- Configure a sus productores para que escriban los datos en los temas de un clúster de la misma AWS región.

Recomendamos crear el replicador con una replicación de nombres de temas idéntica (mantenga el mismo nombre de tema en la consola). Esto simplifica el proceso de conmutación por error, ya que no requiere cambios en la cadena de arranque ni en la reconfiguración del nombre del tema. 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.

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 MSK Replicator los copia automáticamente si ha seleccionado la opción de copiar las listas de control de acceso.

Consulte [Replicación de metadatos y ACL](#).

Replicación activa-activa

En una configuración activo-activo, ambos clústeres de MSK prestan servicios activos de lectura y escritura. Una topología de clúster activo-activo ofrece un tiempo de recuperación prácticamente nulo y la capacidad de que su aplicación de streaming funcione simultáneamente en varias regiones. AWS

Utilizar la replicación de nombres de temas con prefijo (recomendado)

Se recomienda utilizar la replicación de nombres de temas con prefijos (añadir un prefijo al nombre de los temas en la consola) para las configuraciones activo-activas. Esto requiere reconfigurar a los usuarios para que lean los temas replicados, pero evita cargos adicionales por el procesamiento de datos.

1. Cree un replicador de 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, cada uno de los cuales escriba datos al mismo tiempo en el tema local (por ejemplo, «tema») del clúster de la misma región que el productor.
4. Cree dos conjuntos de consumidores, cada uno de los cuales lea los datos mediante una suscripción comodín (por ejemplo, `*topic`) del clúster de MSK de la misma AWS región que el consumidor. De esta forma, sus consumidores leerán automáticamente los datos generados localmente a partir del tema local (por ejemplo, `topic`), así como los datos replicados de la otra región del tema con el prefijo. `<sourceKafkaClusterAlias>.topic` Estos dos grupos de consumidores deben tener un grupo de consumidores diferente IDs para que las compensaciones entre grupos de consumidores no se sobrescriban cuando MSK Replicator las copie en el otro clúster.

Uso de la replicación de nombres de temas idénticos

Si desea evitar la reconfiguración de sus clientes, puede crear los replicadores de MSK mediante la replicación de nombres de temas idénticos (mantenga el mismo nombre de tema en la consola). 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`.

Note

Si utiliza la replicación de nombres de tema idénticos para la topología activo-activa, espere al menos 30 segundos después de eliminar un tema antes de volver a crear uno con el mismo nombre. Este período 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.

Conmutación por error a la región secundaria AWS

Le recomendamos que supervise la latencia de replicación en la AWS región secundaria mediante Amazon CloudWatch. Durante un evento de servicio en la AWS región principal, la latencia de replicación puede aumentar repentinamente. Si la latencia sigue aumentando, usa el [AWS Service Health Dashboard](#) para comprobar si hay eventos de servicio en la AWS región principal. Si se produce un evento de servicio, puedes realizar la conmutación por error a la AWS región secundaria.

Conmutación por error planificada

Puede realizar una conmutación por error planificada para comprobar la resiliencia de su aplicación ante un suceso inesperado en su región principal. AWS Una conmutación por error planificada no debería provocar la pérdida de datos.

Identical topic name replication

1. Cierre todos los productores y consumidores que se conecten a su 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 escritos en la región secundaria y devolverlos a la región principal para su recuperación por recuperación.
3. Inicie la conexión de productores y consumidores con el clúster objetivo en la AWS región secundaria.

Prefixed topic name replication

1. Cierre todos los productores y consumidores que se conecten a su 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 la recuperación por recuperación.
3. Inicie a los productores en el clúster objetivo de la AWS región secundaria.

4. Si su aplicación no requiere ordenar los mensajes, utilice un operador comodín para que los consumidores de la AWS región secundaria lean tanto temas locales (por ejemplo `<sourceKafkaClusterAlias>.topic`) como replicados (por ejemplo `.topic.*topic`).

Si su aplicación requiere ordenar los mensajes, seleccione Consumers solo para los temas replicados en el clúster de destino (por ejemplo `<sourceKafkaClusterAlias>.topic`), pero no para los temas locales.

5. Espere a que todos los consumidores de los temas replicados en el clúster de MSK de destino terminen de procesar todos los datos, de modo que el retraso entre los consumidores sea 0. A continuación, detenga a los consumidores por los temas replicados.
6. Inicie los consumidores para los temas locales (por ejemplo, `topic`) del clúster de MSK de destino.

Conmutación por error no planificada

Puede realizar una conmutación por error no planificada cuando se produzca un evento de servicio en la AWS región principal y desee redirigir temporalmente el tráfico a la región secundaria. 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 [Supervisar la replicación](#).

Identical topic name replication

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 lleve a cabo correctamente debido a problemas en esa región.
2. Haga que los productores y consumidores se conecten con el clúster de MSK objetivo en la región secundaria AWS. Como MSK Replicator también replica los metadatos, incluidas las compensaciones de lectura ACLs y de grupos de consumidores, sus productores y consumidores reanudarán el procesamiento sin problemas prácticamente desde donde lo dejaron.

Prefixed topic name replication

1. Intente desactivar todos los productores y consumidores que se conectan al clúster de MSK de origen de la región principal.

2. Inicie la conexión de los productores con el clúster de MSK de destino en la región secundaria. AWS
3. Si su solicitud no requiere ordenar los mensajes, utilice un operador comodín para que los consumidores que lean tanto los temas locales como los replicados utilicen un comodín. Si su aplicación requiere ordenar los mensajes, inicie primero con solo los temas replicados, espere a que el retraso llegue a 0 y, a continuación, cambie a los temas locales.
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. Si no establece la posición inicial en la primera, los datos generados durante el evento de servicio no se copiarán de nuevo.

Recaída por error

Puede volver por recuperación a la AWS región principal una vez finalizado el evento de servicio en esa región.

Identical topic name replication

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 comienza a 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 de 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 los consumidores que se conecten al clúster secundario se conviertan en MaxOffsetLag métricas 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 su replicador actual que copia datos del clúster principal al secundario tenga el estado «EN EJECUCIÓN» y que la `ReplicatorThroughput` métrica de Amazon CloudWatch sea superior 0 a.

Tenga en cuenta que, al crear un replicador nuevo con la posición inicial de recuperación temprana, empezará a leer todos los datos de los temas del clúster secundario. 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`.

Prefixed topic name replication

Debe iniciar los pasos de recuperación solo después de que la replicación del clúster de la región secundaria al clúster de la región principal se haya puesto al día y la `MessageLag` métrica de Amazon CloudWatch esté cerca de 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 activa-pasiva, elimine el replicador que está replicando 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. Si su aplicación no requiere ordenar los mensajes, utilice un operador comodín para que los consumidores de la AWS región principal lean tanto los temas locales como los replicados. Si su aplicación requiere ordenar los mensajes, inicie primero con solo los temas replicados, espere a que el retraso llegue a 0 y, a continuación, cambie a los temas locales.
5. Compruebe que el replicador existente desde el clúster de la región principal hasta el clúster de la región secundaria esté en ejecución y funcione según lo previsto utilizando las métricas `ReplicatorThroughput` y de latencia.

Migración

MSK Replicator admite varios escenarios de migración. En esta sección, se describe cómo migrar datos entre clústeres, desde entornos Kafka autogestionados a Amazon MSK, y cómo realizar la transición desde otras herramientas de replicación.

Temas

- [Migre entre clústeres de Amazon MSK](#)
- [Migre de clústeres de Apache Kafka que no son de MSK a corredores de Amazon MSK Express](#)
- [Migre de MirrorMaker 2 a Amazon MSK Replicator](#)
- [Cross-account migración entre clústeres de Amazon MSK](#)

Migre entre clústeres de Amazon MSK

Replicador Amazon MSK admite migraciones entre clústeres de Amazon MSK en la misma Cuenta de AWS. 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 una replicación al menos una vez, 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 necesita configurar los ajustes a nivel de tema ni las ACL de lectura «literal», ya que MSK Replicator las copia automáticamente.
3. Supervisa 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.

Migre de clústeres de Apache Kafka que no son de MSK a corredores de Amazon MSK Express

Puede usar MSK Replicator para migrar cargas de trabajo de Apache Kafka desde entornos autogestionados a clústeres aprovisionados de Amazon MSK con agentes Express. MSK Replicator admite la migración de datos desde las implementaciones de Kafka (versión 2.8.1 o posterior de Kafka) que tienen habilitada la autenticación. SASL/SCRAM

Note

SASL/SCRAM La autenticación solo es necesaria para que MSK Replicator se conecte a su clúster Kafka autogestionado. Las aplicaciones cliente pueden seguir utilizando sus mecanismos de autenticación existentes.

Requisitos previos

Antes de comenzar, asegúrese de que dispone de lo siguiente:

1. Clúster Apache Kafka de origen que ejecuta la versión 2.8.1 o posterior
2. SASL/SCRAM autenticación habilitada en el clúster de origen
3. cifrado SSL configurado en el clúster de origen
4. Conectividad de red mediante AWS Site-to-Site VPN o AWS Direct Connect
5. Subredes de VPC configuradas para el acceso a Secrets Manager

Para obtener instrucciones detalladas, consulte [Configure los requisitos previos para MSK Replicator con clústeres de Apache Kafka autogestionados](#).

Paso 1: Crear un clúster de Amazon MSK Express

Cree un clúster aprovisionado por MSK con Express Brokers con la autenticación de IAM habilitada. Mínimo tres corredores en tres zonas de disponibilidad. Consulte [Prepare el clúster de destino](#).

Paso 2: Crear una función de ejecución de IAM

Adjunte la política `AWSMSKReplicatorExecutionRole` gestionada y configure la política de confianza `parakafka.amazonaws.com`. Agregue permisos en línea para AWS Secrets Manager (y AWS KMS si sus secretos lo son CMK-encrypted) per. [Permisos SER adicionales y claves administradas por el SASL/SCRAM cliente](#) Consulte [Configure los requisitos previos para MSK Replicator con clústeres de Apache Kafka autogestionados](#).

Paso 3: Configurar SASL/SCRAM un SSL en un clúster autogestionado

Cree un usuario de SCRAM dedicado con los permisos de ACL necesarios. Configure los certificados SSL. Consulte [Configure los requisitos previos para MSK Replicator con clústeres de Apache Kafka autogestionados](#).

Paso 4: Almacene las credenciales en AWS Secrets Manager

Cree pares secretos con `usernamepassword`, y `certificate` clave-valor. Consulte [Configure los requisitos previos para MSK Replicator con clústeres de Apache Kafka autogestionados](#).

Paso 5: Crea el replicador

Utilice la `CreateReplicator` API con la posición `EARLIEST` inicial, la replicación del nombre del tema idéntica y configúrela `synchroniseConsumerGroupOffsets` `true` en. El principio de IAM al que se llama `CreateReplicator` debe tener los permisos de llamada a la API descritos en. [Permisos de IAM obligatorios para crear un Replicador MSK](#) Si planea configurar la replicación bidireccional para obtener la capacidad de reversión (paso 6), configúrela también `ENHANCED` en `consumerGroupOffsetSyncMode` los replicadores directos e inversos. Espere aproximadamente 30 minutos para que el replicador alcance el estado `EN FUNCIONAMIENTO`. Consulte [CreateReplicator Ejemplos de API para clústeres de Kafka autogestionados](#).

Paso 6: (opcional) Configurar la replicación bidireccional

Cree un replicador inverso desde el clúster de MSK Express hasta el clúster autogestionado para obtener las capacidades de reversión. Consulte [CreateReplicator Ejemplos de API para clústeres de Kafka autogestionados](#).

Paso 7: Supervise el progreso de la replicación

Supervise las siguientes métricas:

- `MessageLag`(debería llegar a 0)
- `ReplicationLatency`

- `ConsumerGroupOffsetSyncFailure`(debería ser 0)
- `ConsumerGroupCount`
- `OffsetLag (MSK Cluster)` y `OffsetLag (Non-MSK Cluster)`

Para obtener más información, consulte [Supervisar la replicación](#).

Paso 8: migrar aplicaciones

Siga estos pasos para migrar sus aplicaciones:

1. Impida que los productores escriban en un clúster autogestionado
2. Reconfigure los productores en el clúster de MSK Express con autenticación de IAM
3. Supervise `MessageLag` hasta que llegue a 0
4. Detenga a los consumidores en un clúster autogestionado
5. Reconfigure a los consumidores en el clúster de MSK Express

Paso 9: (opcional) Volver al clúster autogestionado

Si se configuró la replicación bidireccional, puede revertir los pasos de migración para volver al clúster autogestionado. El replicador inverso (MSK Express → Externo) habrá mantenido sincronizado el clúster autogestionado, de forma que los consumidores puedan volver a ser redirigidos sin pérdida de datos.

Configure los requisitos previos para MSK Replicator con clústeres de Apache Kafka autogestionados

Creación de un rol de ejecución de IAM

Cree un rol de IAM con una política de confianza para `kafka.amazonaws.com`. Adjunte la política `AWSMSKReplicatorExecutionRole` gestionada. La política gestionada concede a Kafka los permisos que Kafka necesita a nivel de clúster, tema y grupo de consumidores, pero no incluye AWS Secrets Manager AWS KMS los permisos necesarios para la autenticación y las credenciales. SASL/SCRAM CMK-encrypted Para ver los fragmentos de políticas en línea que puede agregar, consulte. [Permisos SER adicionales y claves administradas por el SASL/SCRAM cliente](#)

Ejemplo de política de confianza:

```
{
```

```
"Statement": [{
  "Effect": "Allow",
  "Principal": {"Service": "kafka.amazonaws.com"},
  "Action": "sts:AssumeRole"
}]
}
```

Configure SASL/SCRAM los permisos de usuario y ACL

Cree un usuario de SCRAM dedicado en su clúster Kafka autogestionado. Se requieren los siguientes permisos de ACL:

1. Lea y describa todos los temas
2. Lea y describa sobre todos los grupos de consumidores
3. Describa un recurso de clúster

Ejemplos de comandos de kafka-acls.sh:

```
# Grant Read and Describe on all topics
kafka-acls.sh --bootstrap-server <broker>:9092 \
  --add --allow-principal User:msk-replicator \
  --operation Read --operation Describe \
  --topic '*'

# Grant Read and Describe on all consumer groups
kafka-acls.sh --bootstrap-server <broker>:9092 \
  --add --allow-principal User:msk-replicator \
  --operation Read --operation Describe \
  --group '*'

# Grant Describe on cluster
kafka-acls.sh --bootstrap-server <broker>:9092 \
  --add --allow-principal User:msk-replicator \
  --operation Describe --cluster
```

Configure SSL en un clúster autogestionado

Configure los detectores de SSL en sus corredores. En el caso de los certificados de confianza pública, no se requiere ninguna configuración adicional. En el caso de los certificados privados o autofirmados, incluya toda la cadena de certificados de CA en el secreto almacenado en AWS Secrets Manager.

Guarde las credenciales en AWS Secrets Manager

Cree un secreto de tipo Other (not RDS/Redshift) en AWS Secrets Manager con los siguientes pares clave-valor:

1. `username`— Nombre de usuario SCRAM para el clúster autogestionado
2. `password`— Contraseña SCRAM para el clúster autogestionado
3. `certificate`— Cadena de certificados CA (formato PEM; obligatorio para private/self los certificados firmados)

Configuración de la conectividad de red

MSK Replicator requiere conectividad de red con su clúster Kafka autogestionado. Opciones compatibles:

- AWS Site-to-Site VPN: conecta las redes locales a tu VPC a través de Internet.
- AWS Direct Connect: establezca una conexión de red privada dedicada desde sus instalaciones hasta AWS.

Configuración de grupos de seguridad

Asegúrese de que los grupos de seguridad permitan el tráfico entre MSK Replicator y el clúster autogestionado en el puerto SASL_SSL (normalmente el 9096). Actualice las reglas de entrada en los grupos de seguridad de VPC y las reglas de salida en el firewall de clúster autogestionado.

CreateReplicator Ejemplos de API para clústeres de Kafka autogestionados

Replicación directa (de Self-managed Kafka a MSK Express)

Utilice el siguiente comando AWS CLI para crear un replicador que replique los datos de su clúster de Kafka autogestionado en un clúster de Amazon MSK Express.

```
aws kafka create-replicator \
  --replicator-name my-selfmanaged-to-msk-replicator \
  --description "Replicating from self-managed Kafka to MSK Express" \
  --service-execution-role-arn arn:aws:iam::123456789012:role/MSKReplicatorRole \
  --kafka-clusters '[
    {
      "apacheKafkaCluster": {
        "bootstrapBrokerString": "broker1:9094,broker2:9094,broker3:9094",
```



```

    "apacheKafkaClusterId": "<self-managed-cluster-id>"
  },
  "clientAuthentication": {
    "saslScram": {
      "mechanism": "SHA256",
      "secretArn": "arn:aws:secretsmanager:us-east-1:123456789012:secret:kafka-
scram-creds"
    }
  },
  "encryptionInTransit": {
    "encryptionType": "TLS",
    "rootCaCertificate": "arn:aws:secretsmanager:us-
east-1:123456789012:secret:kafka-scram-creds"
  },
  "vpcConfig": {
    "subnetIds": ["subnet-aaa","subnet-bbb","subnet-ccc"],
    "securityGroupIds": ["sg-xxxxxxxxxxxxxxxxxxxx"]
  }
},
{
  "amazonMskCluster": {
    "mskClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/msk-express/xxx"
  },
  "vpcConfig": {
    "subnetIds": ["subnet-ddd","subnet-eee","subnet-fff"],
    "securityGroupIds": ["sg-yyyyyyyyyy"]
  }
}]' \
--replication-info-list '[{
  "sourceKafkaClusterId": "<self-managed-cluster-id>",
  "targetKafkaClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/msk-express/
xxx",
  "targetCompressionType": "NONE",
  "topicReplication": {
    "topicsToReplicate": [".*"],
    "topicNameConfiguration": {"type": "IDENTICAL"},
    "startingPosition": {"type": "EARLIEST"},
    "detectAndCopyNewTopics": true,
    "copyTopicConfigurations": true,
    "copyAccessControlListsForTopics": true
  },
  "consumerGroupReplication": {
    "consumerGroupsToReplicate": [".*"],
    "detectAndCopyNewConsumerGroups": true,

```

```
"synchroniseConsumerGroupOffsets": true
}}]'
```

Ejemplo de replicación bidireccional

Para configurar la replicación bidireccional con la capacidad de reversión, tanto los replicadores directos como los inversos se deben crear con el valor establecido en `consumerGroupOffsetSyncMode ENHANCED`. Esto garantiza que las compensaciones de los grupos de consumidores estén sincronizadas de una manera que permita una transición perfecta en cualquier dirección.

Cree el replicador directo (autogestionado de Kafka a MSK Express) con el modo de sincronización de offset: `ENHANCED`

```
aws kafka create-replicator \
  --replicator-name my-selfmanaged-to-msk-replicator \
  --description "Replicating from self-managed Kafka to MSK Express" \
  --service-execution-role-arn arn:aws:iam::123456789012:role/MSKReplicatorRole \
  --kafka-clusters '[
    {
      "apacheKafkaCluster": {
        "bootstrapBrokerString": "broker1:9094,broker2:9094,broker3:9094",
        "apacheKafkaClusterId": "<self-managed-cluster-id>"
      },
      "clientAuthentication": {
        "saslScram": {
          "mechanism": "SHA256",
          "secretArn": "arn:aws:secretsmanager:us-east-1:123456789012:secret:kafka-
scram-creds"
        }
      },
      "encryptionInTransit": {
        "encryptionType": "TLS",
        "rootCaCertificate": "arn:aws:secretsmanager:us-
east-1:123456789012:secret:kafka-ca-cert"
      },
      "vpcConfig": {
        "subnetIds": ["subnet-aaa","subnet-bbb","subnet-ccc"],
        "securityGroupIds": ["sg-xxxxxxxxxxxxxxxxxxxx"]
      }
    },
    {
      "amazonMskCluster": {
```

```

        "mskClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/msk-express/xxx"
    },
    "vpcConfig": {
        "subnetIds": ["subnet-ddd","subnet-eee","subnet-fff"],
        "securityGroupIds": ["sg-yyyyyyyyyy"]
    }
}]' \
--replication-info-list '[{
    "sourceKafkaClusterId": "<self-managed-cluster-id>",
    "targetKafkaClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/msk-express/
xxx",
    "targetCompressionType": "NONE",
    "topicReplication": {
        "topicsToReplicate": [".*"],
        "topicNameConfiguration": {"type": "IDENTICAL"},
        "startingPosition": {"type": "EARLIEST"},
        "detectAndCopyNewTopics": true,
        "copyTopicConfigurations": true,
        "copyAccessControlListsForTopics": true
    },
    "consumerGroupReplication": {
        "consumerGroupsToReplicate": [".*"],
        "detectAndCopyNewConsumerGroups": true,
        "synchroniseConsumerGroupOffsets": true,
        "consumerGroupOffsetSyncMode": "ENHANCED"
    }
}]'
```

A continuación, cree el replicador inverso (de MSK Express a Kafka autogestionado) también con el modo de sincronización offset: ENHANCED

```

aws kafka create-replicator \
--replicator-name my-msk-to-selfmanaged-replicator \
--description "Reverse replication for rollback" \
--service-execution-role-arn arn:aws:iam::123456789012:role/MSKReplicatorRole \
--kafka-clusters '[
{
    "amazonMskCluster": {
        "mskClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/msk-express/xxx"
    },
    "vpcConfig": {
        "subnetIds": ["subnet-ddd","subnet-eee","subnet-fff"],
        "securityGroupIds": ["sg-yyyyyyyyyy"]
    }
}]'
```

```

    },
    {
      "apacheKafkaCluster": {
        "bootstrapBrokerString": "broker1:9094,broker2:9094,broker3:9094",
        "apacheKafkaClusterId": "<self-managed-cluster-id>"
      },
      "clientAuthentication": {
        "saslScram": {
          "mechanism": "SHA256",
          "secretArn": "arn:aws:secretsmanager:us-east-1:123456789012:secret:kafka-
scram-creds"
        }
      },
      "encryptionInTransit": {
        "encryptionType": "TLS",
        "rootCaCertificate": "arn:aws:secretsmanager:us-
east-1:123456789012:secret:kafka-ca-cert"
      },
      "vpcConfig": {
        "subnetIds": ["subnet-aaa","subnet-bbb","subnet-ccc"],
        "securityGroupIds": ["sg-xxxxxxxxxxxxxxxxxxxx"]
      }
    }
  ]]' \
--replication-info-list '[{
  "sourceKafkaClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/msk-express/
xxx",
  "targetKafkaClusterId": "<self-managed-cluster-id>",
  "targetCompressionType": "NONE",
  "topicReplication": {
    "topicsToReplicate": [".*"],
    "topicNameConfiguration": {"type": "IDENTICAL"},
    "startingPosition": {"type": "LATEST"},
    "detectAndCopyNewTopics": true,
    "copyTopicConfigurations": true,
    "copyAccessControlListsForTopics": true
  },
  "consumerGroupReplication": {
    "consumerGroupsToReplicate": [".*"],
    "detectAndCopyNewConsumerGroups": true,
    "synchroniseConsumerGroupOffsets": true,
    "consumerGroupOffsetSyncMode": "ENHANCED"
  }
}]'
```

Verifique el estado del replicador

Compruebe el estado de su replicador mediante el comando `describe-replicator` CLI:

```
aws kafka describe-replicator \
  --replicator-arn arn:aws:kafka:us-east-1:123456789012:replicator/my-replicator/xxx
```

El replicador avanzará por `CREATING` → `RUNNING` estados. Espere aproximadamente 30 minutos para que el replicador alcance `RUNNING` el estado.

Migre de MirrorMaker 2 a Amazon MSK Replicator

Para migrar de MirrorMaker (MM2) a MSK Replicator, siga estos pasos:

1. Detenga al productor que está escribiendo en su clúster de Amazon MSK de origen.
2. Permita que MM2 replique todos los mensajes de los temas de su clúster de origen. Puede monitorizar el retraso de consumo del consumidor de MM2 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 nombre de tema establecida en `IDENTICAL` (mantenga el mismo nombre de tema 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.

Cross-account migración entre clústeres de Amazon MSK

Para las migraciones entre AWS cuentas diferentes, debe usar Apache MirrorMaker 2.0.

[Apache MirrorMaker 2.0](#) es una herramienta de código abierto que requiere configuración y administración manuales, pero proporciona un control detallado del proceso de migración. Para obtener más información, consulte la documentación de [georreplicación de Kafka](#).

Conceptos clave

En esta sección, se describen los conceptos clave de MSK Replicator, como la forma en que los datos fluyen entre los clústeres, la forma en que se gestionan los nombres de los temas y la forma en que se replican los metadatos.

Temas

- [Cómo funciona la replicación](#)
- [Denominación de temas \(prefijo o idéntica\)](#)
- [Replicación de metadatos y ACL](#)
- [Sincronización compensada por grupos de consumidores](#)
- [Requisitos de red](#)

Cómo funciona la replicación

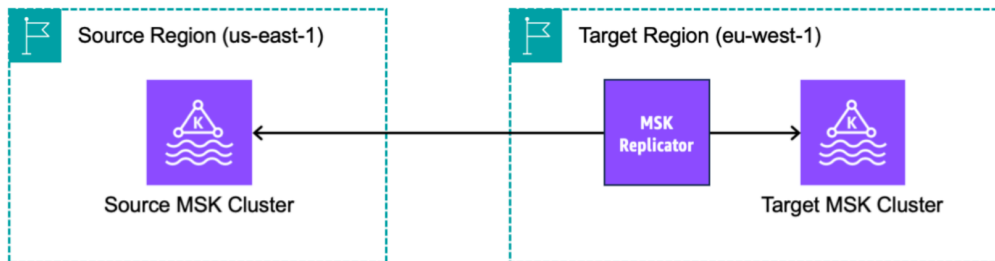
Al crear un replicador, MSK Replicator despliega todos los recursos necesarios en la AWS región del clúster de destino para optimizar la latencia de la replicación de datos. MSK Replicator copia automáticamente todos los datos del clúster de la AWS región principal denominada origen al clúster de la región de destino denominada destino. Los clústeres de origen y destino pueden estar en la misma región o en regiones diferentes AWS .

La latencia de replicación varía en función de muchos factores, como la distancia de red entre las AWS regiones de los clústeres de MSK, la capacidad de rendimiento de los clústeres de origen y destino y el número 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.

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 su replicador durante la creación para iniciar la replicación desde el primer momento si desea replicar los mensajes existentes.

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. Todas las comunicaciones y los datos entre MSK Replicator y sus clústeres siempre se cifran durante el tránsito. Todas las llamadas a la API de MSK Replicator `DescribeClusterV2`, como, `CreateTopic` se capturan. `DescribeTopicDynamicConfiguration` AWS CloudTrail

MSK Replicator 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.



Denominación de temas (prefijo o idéntica)

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 distingue los temas replicados de otros del clúster de destino y evita la replicación circular de datos entre los clústeres.

Por ejemplo, MSK Replicator replica los datos de un tema denominado «tema» del clúster de origen a un tema nuevo del clúster de destino denominado `<sourceKafkaClusterAlias>.topic`. Puedes encontrar el prefijo en el campo `sourceKafkaClusterAlias` mediante la `DescribeReplicator` API o la página de detalles del replicador en la consola de MSK.

Para asegurarse de que sus consumidores puedan reiniciar el procesamiento de forma fiable desde el clúster en espera, configúrelos para que lean los datos de los temas mediante un operador comodín. `.*` Por ejemplo, sus consumidores deberían consumir `.topic1` en ambas AWS regiones. Este ejemplo también incluiría un tema como `footopic1`, así que ajuste el operador comodín según sus necesidades.

Utilice la replicación de nombres de temas con prefijo cuando desee mantener los datos replicados en un tema independiente del clúster de destino, por ejemplo, en el caso de configuraciones de clústeres activo-activos. En la configuración con prefijo, ambos `ReplicatorBytesInPerSec` `ReplicatorThroughput` tendrán el mismo valor, ya que MSK Replicator no filtrará ningún dato.

Replicación de nombre de tema idéntico

Como alternativa, Amazon MSK Replicator le permite crear un replicador con la replicación de temas establecida en la replicación de nombres de temas idénticos (mantenga el mismo nombre de tema en la consola). 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 tiene las siguientes ventajas:

- Conserva los nombres de temas idénticos durante la replicación y, al mismo tiempo, evita automáticamente los bucles de replicación infinitos.
- Simplifica las arquitecturas de transmisión de varios clústeres, ya que puede evitar la reconfiguración de los clientes.
- Simplifica el proceso de conmutación por error para las arquitecturas activas y pasivas, lo que permite a las aplicaciones realizar la conmutación por error sin necesidad de cambiar el nombre de los temas ni reconfigurar los clientes.
- Puede consolidar los datos de varios clústeres de MSK en un solo clúster para la agregación de datos o el análisis centralizado (requiere replicadores independientes para cada clúster de origen).
- Puede agilizar la migración de datos de un clúster de MSK a otro.

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. MSK Replicator incorpora los identificadores del clúster y el tema de origen en el encabezado de cada registro que se está replicando (`__mskmr`). Debe comprobar que sus clientes pueden leer los datos replicados según lo esperado.

Replicación de metadatos y ACL

MSK Replicator permite copiar 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 las compensaciones de los grupos de consumidores. Al igual que la replicación de datos, la replicación de metadatos se produce de forma asíncrona. Para mejorar el rendimiento, el Replicador MSK prioriza la replicación de datos sobre la replicación de metadatos.

En la siguiente tabla se enumeran las listas de control de acceso (ACLs) que copia MSK Replicator.

Operación	Recurso	APIs permitido
Modificación	Topic	CreatePartitions
AlterConfigs	Topic	AlterConfigs
Crear	Topic	CreateTopics, Metadatos
Eliminar	Topic	DeleteRecords, DeleteTopics
Describir	Topic	ListOffsets, Metadatos OffsetFetch, OffsetFor LeaderEpoch
DescribeConfigs	Topic	DescribeConfigs
Lectura	Topic	Buscar,, OffsetCommit TxnOffsetCommit
Escribir (solo denegar)	Topic	Producir, AddPartitionsToTxn

MSK Replicator copia el tipo de patrón LITERAL ACLs solo para el tipo de recurso Topic. El tipo de patrón con prefijo ACLs y otros tipos de recursos ACLs no se copian. MSK Replicator tampoco elimina ACLs en el clúster de destino. Si elimina una ACL del clúster de origen, también debe eliminarla del clúster de destino al mismo tiempo. Para obtener más información sobre Kafka ACLs, consulte la documentación de la [ACL de Kafka](#).

MSK Replicator solo replica a Kafka ACLs, que el control de acceso de IAM no utiliza. Si sus clientes utilizan el control de acceso de IAM read/write a 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 es válido tanto para las configuraciones de replicación de nombres de temas con prefijo como para las idénticas.

Note

MSK Replicator no replica la escritura, ACLs ya que sus 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.

Sincronización compensada por grupos de consumidores

MSK Replicator puede sincronizar las compensaciones de grupos de consumidores de un clúster de origen a un clúster de destino, lo que permite a los consumidores cambiar de clúster y reanudar el procesamiento sin omitir registros. En este tema se explica cómo funciona la sincronización de compensaciones en configuraciones unidireccionales (antiguas) y bidireccionales (mejoradas), y se destacan los errores más comunes.

Cómo funciona la sincronización de offset

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. Si activó la sincronización de compensaciones de grupos de consumidores durante la creación de Replicator, MSK Replicator traduce automáticamente las compensaciones mientras copia los metadatos para que, tras la conmutación por error al clúster de destino, sus consumidores puedan reanudar el procesamiento prácticamente desde donde lo dejaron.

MSK Replicator se optimiza para los consumidores del clúster de origen que leen desde una posición más cercana a la punta de la secuencia (al final de la partición del tema). Si tus grupos de consumidores están rezagados en el clúster de origen, es posible que veas un retraso mayor en los grupos de consumidores del grupo de destino que en el de origen, lo que significa que los consumidores volverán a procesar más mensajes duplicados tras la conmutación por error. Para reducir este retraso, los consumidores del clúster de origen deben ponerse al día y empezar a consumir desde el principio. A medida que sus consumidores se pongan al día, el Replicador MSK reducirá automáticamente el retraso.

La sincronización Offset se realiza en tres etapas:

1. Mapeo de desplazamiento: a medida que los registros se replican del origen al destino, el replicador registra mapeos periódicos entre los desplazamientos de origen y los desplazamientos de destino correspondientes. Como las compensaciones de origen y destino son divergentes (diferentes puntos de partida, compactación, etc.), estas asignaciones son esenciales.
2. Traducción de compensaciones: periódicamente, el replicador lee las compensaciones confirmadas para cada grupo de consumidores replicado en el clúster de origen. A continuación, utiliza las asignaciones de desplazamiento almacenadas para convertir esas compensaciones de origen en las compensaciones de destino equivalentes.

3. Compensación de compensación: las compensaciones traducidas se asignan al `__consumer_offsets` tema del clúster de destino, de modo que cuando un consumidor se conecta al objetivo y se une al mismo grupo, se retoma aproximadamente desde la posición correcta.

Comportamientos clave:

- La traducción offset es aproximada, no exacta. El replicador toma muestras de los mapeos de desplazamiento a intervalos, por lo que el desplazamiento traducido puede estar ligeramente por detrás de la posición equivalente real. Esto se debe a su diseño: se equivoca en el lado de la at-least-once entrega, lo que significa que los consumidores pueden volver a leer una pequeña cantidad de mensajes tras la conmutación por error.
- Las compensaciones solo se asignan a un grupo de consumidores del objetivo si ningún consumidor consume activamente en ese grupo del objetivo. Esto evita que el replicador interfiera con los consumidores que ya utilizan el clúster de destino.
- Los grupos de consumidores deben coincidir con el filtro de grupos de consumidores configurado (patrones de inclusión/exclusión) para poder sincronizarse.

Replicación unidireccional con sincronización de offset tradicional

Este es el modo predeterminado para un replicador unidireccional estándar (Clúster A → Clúster B).

- Denominación de temas: la sincronización de offset tradicional admite la replicación de nombres de temas con prefijo e idénticos.
- Dirección: las compensaciones de grupos de consumidores solo se sincronizan con el clúster de destino cuando los productores están activos en el grupo de origen y los consumidores están inactivos en el grupo de destino.
- Conmutación por error: se puede dirigir a los consumidores al clúster de destino y se reanudarán desde la posición de compensación traducida.
- No admite la conmutación por recuperación: la sincronización de compensaciones antigua no traduce las compensaciones del destino a la fuente. Si trasladas a los consumidores al destino y luego quieres devolverlos al lugar de origen, no habrá traducción automática en offset para el viaje de ida y vuelta. Si se requiere la conmutación por recuperación, utilice una configuración bidireccional con una sincronización de offset mejorada.

Configuración bidireccional con sincronización de offset mejorada

Una configuración bidireccional requiere dos replicadores que funcionen en direcciones opuestas (Replicador A→B y Replicador B→A). Cada replicador sigue realizando la replicación de datos unidireccional y la sincronización de compensaciones: replica los datos desde su origen hasta su destino y sincroniza las compensaciones de los grupos de consumidores en la misma dirección. Gracias a la sincronización de compensaciones mejorada, cada replicador puede seguir sincronizando los grupos de consumidores incluso cuando los productores y los consumidores estén activos en clústeres diferentes.

Características clave:

- Denominación de temas: la sincronización mejorada requiere la replicación de nombres de temas idénticos en ambos replicadores.
- Dos replicadores, cada uno unidireccional: cada replicador replica los datos y sincroniza las compensaciones en una dirección. El comportamiento bidireccional proviene del trabajo conjunto de ambos.
- Lee los mapeos de ambos replicadores: al traducir las compensaciones, los replicadores trabajan juntos para determinar la traducción más precisa disponible.
- Conmutación por error y recuperación: los consumidores se pueden mover de un clúster a otro y volver a utilizarlos aproximadamente desde la posición correcta.

Cuándo habilitar la sincronización de compensación bidireccional:

- Migración con capacidad de reversión: cuando se migra de un clúster a otro y se desea tener la posibilidad de volver al clúster original en caso de que surjan problemas.
- Arquitecturas activas-activas: cuando ambos clústeres prestan servicios de lectura y escritura de forma activa y se necesita que los consumidores puedan cambiar de un clúster a otro.
- Recuperación ante desastres: cuando se necesita garantizar que los consumidores puedan reanudar el procesamiento desde la posición correcta en cualquiera de los clústeres tras una conmutación por error o una conmutación por recuperación.

Supervisión de la sincronización de la compensación

Supervisa las siguientes CloudWatch métricas de Amazon para comprobar que la sincronización de compensaciones funciona correctamente:

- `ConsumerGroupCount`— Compruebe que el número esperado de grupos de consumidores se esté sincronizando en ambos replicadores.
- `ConsumerGroupOffsetSyncFailure`— Debe ser 0 en ambos replicadores. Si este valor es superior a 0, compruebe que los grupos de consumidores estén activos, compruebe los permisos de lectura y descripción y asegúrese de que los temas existan en el clúster de destino.
- `OffsetLag (MSK Cluster)` y `OffsetLag (Non-MSK Cluster)`: compare el desfase de consumo a nivel de partición en ambos clústeres para comprobar que las compensaciones estén sincronizadas.

Dificultades comunes

1. Los consumidores pueden volver a leer una pequeña cantidad de mensajes tras la conmutación por error

La traducción offset es aproximada. El desfase traducido es intencionadamente conservador; es posible que esté ligeramente por detrás de la posición equivalente real. Esto significa que los consumidores suelen volver a procesar una pequeña cantidad de registros después de cambiar de clúster. Las aplicaciones deben diseñarse para gestionar el procesamiento duplicado (idempotencia).

2. Las compensaciones no se sincronizan con los grupos que consumen energía de forma activa en el objetivo

Si un grupo de consumidores ya está activo en el clúster de destino, el replicador no sobrescribirá sus compensaciones. Se trata de un mecanismo de seguridad. Sin embargo, esto significa que si los consumidores comienzan con el objetivo antes de que el replicador haya tenido la oportunidad de sincronizar las compensaciones, esos consumidores comenzarán con la política de restablecimiento de compensaciones predeterminada del objetivo (normalmente `latest` o `earliest`), no desde la posición traducida.

3. La sincronización de compensaciones tiene un retraso inherente

La traducción de offset depende de dos procesos asíncronos: la replicación de datos y la sincronización de offset. Siempre hay algún retraso entre el momento en que un consumidor comete una compensación en la fuente y el momento en que la compensación traducida aparece en la de destino. Durante la conmutación por error, este retraso puede provocar que los consumidores releen más mensajes de los esperados si el consumidor de origen ha estado activo recientemente.

4. Los grupos de consumidores deben incluirse en el filtro de replicación

Solo los grupos de consumidores que coincidan con el patrón de inclusión configurado (y que no coincidan con el patrón de exclusión) tendrán sus compensaciones sincronizadas. Si las compensaciones de un grupo de consumidores no aparecen en el destino, compruebe que estén incluidas en la configuración de replicación del grupo de consumidores.

5. Los replicadores unidireccionales no admiten la conmutación por recuperación

Con la sincronización de compensaciones tradicional (unidireccional), las compensaciones solo se traducen del origen al destino. Si traslada a los consumidores al destino y, posteriormente, necesita volver a colocarlos en el lugar de origen, tendrá que determinar manualmente las compensaciones correctas o aceptar el reprocesamiento. Si se requiere la conmutación por recuperación, utilice una configuración bidireccional con una sincronización de offset mejorada.

6. La eliminación y recreación de temas pueden invalidar los mapeos de offset

Si se elimina un tema y se vuelve a crear en cualquiera de los clústeres, las asignaciones de desplazamiento se vuelven obsoletas porque el nuevo tema comienza desde el desplazamiento 0. Con la sincronización de offset tradicional, esto puede provocar traducciones de offset incorrectas. La sincronización de offset mejorada detecta la recreación de temas y restablece los mapeos automáticamente.

Requisitos de red

MSK Replicator utiliza el control de acceso IAM para conectarse a sus clústeres, que utiliza el puerto 9098. Los requisitos de red varían en función de si está configurando la replicación entre regiones o en la misma región.

Replicación entre regiones (CRR)

El clúster de origen debe tener activada la conectividad privada de varias VPC para el control de acceso de IAM. También debes adjuntar al clúster de origen una política de permisos basada en los recursos para que el director del `kafka.amazonaws.com` servicio pueda realizar y realizar acciones `kafka:CreateVpcConnection`, `kafka:GetBootstrapBrokers`, `kafka:DescribeClusterV2`. No es necesario proporcionar grupos de seguridad para el clúster de origen. Debe proporcionar grupos de seguridad para el clúster de destino con las reglas de entrada y salida adecuadas en el puerto 9098.

Replicación en la misma región (SRR)

No se requiere conectividad privada de múltiples VPC. Debe proporcionar grupos de seguridad para los clústeres de origen y de destino. Asegúrese de que los grupos de seguridad que proporcione para el Replicator tengan reglas de salida que permitan el tráfico a los grupos de seguridad del clúster en el puerto 9098, y de que los grupos de seguridad del clúster tengan reglas de entrada que acepten el tráfico de los grupos de seguridad del Replicator en el puerto 9098. Las subredes que seleccione para los clústeres de origen y destino deben estar en las mismas zonas de disponibilidad.

Tanto para el CRR como para el SRR, asegúrese de que la red ACLs no bloquee la conexión entre el MSK Replicator y los clústeres de origen y destino.

Seguridad

En esta sección se describe el modelo de seguridad de MSK Replicator, incluida la función de ejecución del servicio, el cifrado en tránsito y la referencia a los permisos de IAM.

Temas

- [Función de ejecución de servicios \(SER\)](#)
- [Cifrado](#)
- [Permisos de IAM obligatorios para crear un Replicador MSK](#)
- [Referencia de permisos de IAM](#)

Función de ejecución de servicios (SER)

MSK Replicator utiliza una función de ejecución de servicios (SER) para leer desde el clúster de origen y escribir en el clúster de destino. Debe especificar este rol al crear el replicador.

Puede dejar que la consola de MSK cree el rol automáticamente o proporcionar su propio rol de IAM. Si proporciona su propia función, le recomendamos que le adjunte la política de IAM [AWSMSKReplicatorExecutionRole](#) gestionada.

La función de ejecución del servicio debe tener una política de confianza que permita al director del kafka.amazonaws.com servicio asumir la función. A continuación, se muestra un ejemplo de una política de confianza. <yourAccountID>Sustitúyalo por tu ID de cuenta real.

```
{
  "Statement": [
    {
      "Effect": "Allow",
```

```

    "Principal": {
      "Service": "kafka.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "<yourAccountID>"
      }
    }
  }
]
}

```

Si desea restringir los permisos `kafka-cluster:WriteData`, consulte la sección [Create authorization policies de How IAM access control for Amazon MSK works](#). Debe añadir `kafka-cluster:WriteDataIdempotently` permisos tanto al clúster de origen como al de destino.

Si reutilizas una función de ejecución de servicios entre varios replicadores de MSK, todos estarán sujetos a las mismas cuotas de Kafka. Si quiere mantener cuotas independientes por replicador, utilice roles de ejecución de servicios independientes.

Permisos SER adicionales y claves administradas por el SASL/SCRAM cliente

La política `AWSMSKReplicatorExecutionRole` gestionada cubre los permisos de clústeres, temas y grupos de consumidores para la autenticación de IAM. Cuando realiza replicaciones hacia o desde un clúster que utiliza SASL/SCRAM autenticación (por ejemplo, al migrar desde un clúster autogestionado de Apache Kafka), o cuando su certificado de CA privado o secreto de SCRAM está cifrado con una clave gestionada por el cliente (CMK), debe adjuntar permisos adicionales integrados a la función de ejecución del servicio.

Utilice los siguientes fragmentos además de la política gestionada. Elija el escenario que se adapte a su configuración.

SASL/SCRAM secreto (con o sin el secreto de la CA raíz de TLS)

Otorga al SER permiso para leer las credenciales de SCRAM y (opcionalmente) el certificado de CA privado desde `AWS Secrets Manager` `<saslSecretArn>` Sustitúyalo por el ARN secreto de SCRAM `<privateCaCertSecretArn>` y por el secreto que contiene el certificado de CA (omite el segundo ARN si utiliza un certificado de confianza pública).

```
{
```



```

"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "SecretsManagerPermissions",
    "Effect": "Allow",
    "Action": [
      "secretsmanager:GetResourcePolicy",
      "secretsmanager:GetSecretValue",
      "secretsmanager:DescribeSecret",
      "secretsmanager:ListSecretVersionIds"
    ],
    "Resource": [
      "<saslSecretArn>",
      "<privateCaCertSecretArn>"
    ]
  }
]
}

```

El secreto de SCRAM o el certificado de CA están cifrados con una clave gestionada por el cliente

Si el secreto o el certificado se cifra con una CMK en lugar de con la clave AWS administrada, escríbalo también `kms:Decrypt` en la CMK. `<customerManagedKeyArn>` Sustitúyalo por el ARN CMK.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SecretsManagerPermissions",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetResourcePolicy",
        "secretsmanager:GetSecretValue",
        "secretsmanager:DescribeSecret",
        "secretsmanager:ListSecretVersionIds"
      ],
      "Resource": [
        "<saslSecretArn>",
        "<privateCaCertSecretArn>"
      ]
    },
    {

```

```
        "Sid": "KmsPermissions",
        "Effect": "Allow",
        "Action": "kms:Decrypt",
        "Resource": [
            "<customerManagedKeyArn>"
        ]
    }
}
```

Note

Si prefiere un ámbito más amplio y coherente con los [permisos del proveedor de configuración](#) de MSK Connect, puede utilizarlos `arn:aws:secretsmanager:<region>:<accountID>:secret:AmazonMSK_*` como patrón de recursos en lugar de los ARN secretos individuales.

Cifrado

Toda la comunicación y los datos entre MSK Replicator y sus clústeres siempre se cifran durante el tránsito. MSK Replicator se conecta a sus clústeres mediante el control de acceso IAM en el puerto 9098, que requiere el cifrado TLS.

MSK Replicator no almacena sus datos en reposo. 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.

Permisos de IAM obligatorios para crear un Replicador MSK

El principal de IAM (usuario o rol) al que llama `CreateReplicator` necesita los permisos que se describen en esta sección. Adjunte esta política a la identidad de IAM que corresponda a su cliente. Para obtener orientación general sobre la creación de políticas de autorización, consulte [Crear políticas de autorización](#).

Comience con la política básica que aparece a continuación. Si también configura la entrega de registros, añada el fragmento a cada destino que utilice (consulte). [Permisos adicionales para la entrega de registros](#) Para ver los escenarios de migración autogestionada de Apache Kafka, consulte

la guía adicional sobre las funciones de ejecución de servicios en. [Migre de clústeres de Apache Kafka que no son de MSK a corredores de Amazon MSK Express](#)

Política básica de IAM

Sustituya los marcadores de posición por su ID de cuenta Región de AWS, nombre de la función de ejecución del servicio y los ARN de los clústeres de origen y destino. La acción solo `kafka:TagResource` es necesaria si proporciona etiquetas durante la creación.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MSKReplicatorIAMPassRole",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::<accountID>:role/<serviceExecutionRoleName>",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "kafka.amazonaws.com"
        }
      }
    },
    {
      "Sid": "MSKReplicatorServiceLinkedRole",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::<accountID>:role/aws-service-role/
kafka.amazonaws.com/AWSServiceRoleForKafka*"
    },
    {
      "Sid": "MSKReplicatorActions",
      "Effect": "Allow",
      "Action": [
        "kafka:CreateReplicator",
        "kafka:DescribeReplicator",
        "kafka>DeleteReplicator",
        "kafka:ListReplicators",
        "kafka:ListTagsForResource",
        "kafka:UpdateReplicationInfo",
        "kafka:TagResource"
      ],
      "Resource": [
```

```

        "arn:aws:kafka:<region>:<accountID>:replicator/*"
    ],
    },
    {
        "Sid": "MSKReplicatorListActions",
        "Effect": "Allow",
        "Action": [
            "kafka:ListReplicators"
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Sid": "EC2Actions",
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups",
            "ec2:DescribeVpcs"
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Sid": "MSKClusterActions",
        "Effect": "Allow",
        "Action": [
            "kafka:GetBootstrapBrokers",
            "kafka:DescribeClusterV2"
        ],
        "Resource": [
            "<sourceClusterArn>",
            "<targetClusterArn>"
        ]
    }
]
}

```

Note

Las `ec2:DescribeVpcs` acciones

`ec2:DescribeSubnet` `ec2:DescribeSecurityGroups`, y no admiten permisos a nivel de recurso, por lo que debe especificarlos. "Resource": "*" Consulte las [claves de condición, recursos y acciones para obtener información de referencia sobre Amazon EC2](#).

Permisos adicionales para la entrega de registros

Si configura la entrega de registros en el replicador, añada las instrucciones correspondientes que aparecen a continuación a la política base. Solo necesita los fragmentos de los destinos que active.

Destino CloudWatch de Amazon Logs

Añada la siguiente declaración cuando `cloudWatchLogs.enabled` esté `true` en la `logDelivery` configuración.

```
{
  "Sid": "CloudWatchLogsLogDeliveryActions",
  "Effect": "Allow",
  "Action": [
    "logs:CreateLogDelivery",
    "logs:PutResourcePolicy",
    "logs:DescribeResourcePolicies",
    "logs:DescribeLogGroups",
    "logs:ListLogDeliveries"
  ],
  "Resource": [
    "*"
  ]
}
```

Destino de Amazon S3

Añada las siguientes declaraciones cuando `s3.enabled` esté `true`. Sustituya `<logBucketName>` con el nombre del bucket de destino.

```
[
  {
    "Sid": "S3LogDeliveryActions",
```

```

    "Effect": "Allow",
    "Action": [
        "logs:CreateLogDelivery",
        "logs:ListLogDeliveries"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Sid": "S3BucketLogDeliveryActions",
    "Effect": "Allow",
    "Action": [
        "s3:GetBucketPolicy",
        "s3:PutBucketPolicy"
    ],
    "Resource": "arn:aws:s3:::<logBucketName>"
}
]

```

Destino Firehose

Añada las siguientes instrucciones cuando esté `firehose.enabled: true`
 <accountID>Sustitúyalo por tu Cuenta de AWS ID.

```

[
  {
    "Sid": "FirehoseLogDeliveryActions",
    "Effect": "Allow",
    "Action": [
        "logs:CreateLogDelivery",
        "logs:ListLogDeliveries",
        "firehose:TagDeliveryStream"
    ],
    "Resource": [
        "*"
    ]
  },
  {
    "Sid": "FirehoseLogDeliveryServiceLinkedRole",
    "Effect": "Allow",
    "Action": [
        "iam:CreateServiceLinkedRole"
    ]
  }
]

```

```

    ],
    "Resource": "arn:aws:iam::<accountID>:role/aws-service-role/
delivery.logs.amazonaws.com/AWSServiceRoleForLogDelivery"
  }
]

```

Para obtener más información sobre los permisos de vendes-logs, consulta Cómo [habilitar el registro desde AWS](#) los servicios.

Referencia de permisos de IAM

En la siguiente tabla se resumen las acciones de IAM que necesita la persona que llama a la API (el director de IAM que llama `CreateReplicator` y otras API del replicador). Para ver ejemplos completos de políticas, consulte. [Permisos de IAM obligatorios para crear un Replicador MSK](#)

Action	Description (Descripción)
<code>kafka:CreateReplicator</code>	Otorga permiso para crear un replicador.
<code>kafka:DescribeReplicator</code>	Otorga permiso para describir un replicador.
<code>kafka:UpdateReplicationInfo</code>	Otorga permiso para actualizar la información de replicación de un replicador.
<code>kafka>DeleteReplicator</code>	Otorga permiso para eliminar un replicador.
<code>kafka:ListReplicators</code>	Otorga permiso para enumerar los replicadores.
<code>kafka:TagResource</code>	Otorga permiso para etiquetar a un replicador. Solo es necesario si se proporcionan etiquetas durante la creación.
<code>kafka:ListTagsForResource</code>	Otorga permiso para enumerar las etiquetas de un replicador.

Action	Description (Descripción)
kafka:GetBootstrapBrokers	Otorga permiso para recuperar los puntos finales de Bootstrap Broker para los clústeres de origen y destino durante la creación del replicador.
kafka:DescribeClusterV2	Otorga permiso para describir los clústeres de origen y destino durante la creación del replicador.
iam:PassRole	Otorga permiso para transferir la función de ejecución del servicio <code>akafka.amazonaws.com</code> .
iam:CreateServiceLinkedRole	Otorga permiso para crear el rol <code>AWSServiceRoleForKafka*</code> vinculado al servicio la primera vez que lo use.
ec2:DescribeSubnets, ec2:DescribeSecurityGroups, ec2:DescribeVpcs	Otorga permiso para validar la configuración de VPC proporcionada al replicador.

Para obtener información sobre los permisos de las funciones de ejecución de servicios, consulte la política [AWSMSKReplicatorExecutionRole](#) administrada. Para conocer los escenarios clave gestionados por el cliente SASL/SCRAM y los escenarios clave, consulte [Permisos SER adicionales y claves administradas por el SASL/SCRAM cliente](#).

Precios

Los precios de MSK Replicator se basan en la cantidad de datos replicados. Usted paga por los datos procesados por el replicador. En el caso de la replicación entre regiones, también paga por la transferencia de datos entre regiones.

Si utiliza la replicación de nombres de tema idénticos para las configuraciones activo-activas, cada replicador procesará el doble de la cantidad de datos habitual (una vez para la replicación y otra para filtrarlos y evitar bucles infinitos). Puede realizar un seguimiento del total de datos procesados por cada replicador mediante la métrica `ReplicatorBytesInPerSec`. Consulte [Referencia de métricas](#).

Para obtener información sobre los precios actuales, consulta los [precios de Amazon MSK](#).

Para conocer las cuotas y los límites del servicio de MSK Replicator, consulte las cuotas de [MSK Replicator en](#) la página de cuotas de Amazon MSK.

Observabilidad

En esta sección se describe cómo monitorizar MSK Replicator mediante CloudWatch las métricas de Amazon y cómo solucionar problemas comunes.

Temas

- [Referencia de métricas](#)
- [Supervise la replicación desde clústeres Kafka autogestionados](#)
- [Registros de MSK Replicator](#)

Referencia de métricas

Las siguientes métricas describen el rendimiento o conexión del Replicador MSK.

AuthError las métricas no cubren los errores de autenticación a nivel de tema. Para monitorear los errores de autenticación a nivel de tema de su MSK Replicator, supervise las métricas de Replicator y las métricas a nivel de tema del clúster de origen,. ReplicationLatency MessagesInPerSec Si un tema ReplicationLatency se reduce a 0 pero el tema aún contiene datos que se están generando, esto indica que el Replicator tiene un problema de autenticación con el tema. Compruebe que la función de IAM de ejecución de servicios del replicador tenga los permisos suficientes para acceder al tema.

Tipo de métrica	Métrica	Description (Descripción)	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
Desempeño	ReplicationLatency	Tiempo que tardan en replicarse los registros desde el	ReplicatorName	Milisegundos	Partición	Máximo
			ReplicatorName, Tema	Milisegundos	Partición	Máximo

Tipo de métrica	Métrica	Description (Descripción)	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
		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 soportar la replicación. Se puede producir una latencia de replicación alta cuando				

Tipo de métrica	Métrica	Description (Descripción)	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
		el recuento de particiones es demasiado bajo para un rendimiento alto.				

Tipo de métrica	Métrica	Description (Descripción)	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
Desen	MessageLag	Supervisa la sincronización entre el MSK Replicator 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	ReplicatorName	Recuento	Partición	Sum
			ReplicatorName, Tema	Recuento	Partición	Sum

Tipo de métrica	Métrica	Description (Descripción)	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
		disponible/ se interrumpe, el Replicador terminará de escribir el mensaje que ha consumido 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 que se puede supervisar hasta que el número de mensajes sea 0, lo				

Tipo de métrica	Métrica	Description (Descripción)	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
		que indica que el replicador ha alcanzado el clúster de origen.				

Tipo de métrica	Métrica	Description (Descripción)	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
Desen	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	ReplicatorName	BytesPerSecond	ReplicatorName	Sum

Tipo de métrica	Métrica	Description (Descripción)	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
		configurado con nombres de temas idénticos) para evitar que los datos se copien de nuevo al mismo tema del que se originaron. Si su Replicador está configurado con nombres de temas con "Prefijo", ambas métricas ReplicadorBytesInPerSec y ReplicadorThroughput tendrán el mismo valor, ya				

Tipo de métrica	Métrica	Description (Descripción)	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
Desempeño	ReplicatorThroughput	El número medio de bytes replicados por segundo. Si se ReplicatorThroughput pierde por un tema, compruebe AuthError las métricas para asegurarse de que el replicador pueda comunicarse con los clústeres KafkaClusterPingSuccessCount y, a continuación, compruebe las métricas del clúster	ReplicatorName	BytesPerSecond	Partición	Sum
			ReplicatorName, Tema	BytesPerSecond	Partición	Sum

Tipo de métrica	Métrica	Description (Descripción)	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
		para asegurarse de que el clúster no esté inactivo.				

Tipo de métrica	Métrica	Description (Descripción)	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
Desempeño	ReplicationFailures	Número de errores de replicación. Debe ser 0 para una replicación correcta. Un valor distinto de cero puede indicar limitaciones en el tamaño de los mensajes, infracciones de las marcas horarias o problemas con el tamaño de los lotes de registros.	ReplicationName	Recuento		Sum

Tipo de métrica	Métrica	Description (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	ReplicaName, ClusterAlias	Recuento	Entorno de trabajo	Sum

Tipo de métrica	Métrica	Description (Descripción)	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
		ClusterAl ias, puede identificar si el clúster de origen o de destino presenta errores de autentica ción.				

Tipo de métrica	Métrica	Description (Descripción)	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
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 replicationThroughput es el esperado, la limitación funciona según lo esperado. Si	ReplicationName, ClusterAlias	Milisegundos	Entorno de trabajo	Máximo

Tipo de métrica	Métrica	Description (Descripción)	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
		esta métrica es superior a 0, puede ajustar la limitación en consecuencia.				
Debug	ReplicatorFailure	Número de errores que sufre el replicador.	ReplicatorName	Recuento		Sum

Tipo de métrica	Métrica	Description (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	ReplicatorName, ClusterAlias	Recuento		Sum

Tipo de métrica	Métrica	Description (Descripción)	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
		de Kafka. Según la ClusterAlias dimensión , puede identificar si esta métrica es para el clúster de origen o de destino.				
Grupo de consumidores	ConsumerGroupCount	Número de grupos de consumidores que se están sincronizando. Compruebe que coincide con los grupos de consumidores esperados.	ReplicatorName	Recuento		Sum

Tipo de métrica	Métrica	Description (Descripción)	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
Grupo de consumidores	ConsumerGroupOffsetSyncFailure	Número de errores de sincronización de compensaciones entre grupos de consumidores. Debe ser 0. Si es mayor que 0, compruebe que los grupos de consumidores estén activos y compruebe los permisos.	ReplicatorName	Recuento		Sum

Tipo de métrica	Métrica	Description (Descripción)	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
Grupo de consumidores	OffsetLag (Clúster MSK)	Retraso de consumo a nivel de partición en el clúster de destino de MSK. Compárelo con OffsetLag (un clúster que no es de MSK) para comprobar que el retraso es igual.	Partición	Recuento		Sum

Tipo de métrica	Métrica	Description (Descripción)	Dimensiones	Unidad	Granularidad de métrica sin procesar	Estadística de agregación de métrica sin procesar
Grupo de consumidores	OffsetLag (Clúster que no es de MSK)	Retraso de consumo a nivel de partición en el clúster de origen autogestionado (que no es de MSK). Compárelo con (MSK Cluster OffsetLag).	Partición	Recuento		Sum

Supervise la replicación desde clústeres Kafka autogestionados

Tras crear un replicador de MSK para replicar los datos de un clúster de Apache Kafka autogestionado a los agentes de Amazon MSK Express, supervise las siguientes CloudWatch métricas de Amazon para realizar un seguimiento del progreso de la replicación y determinar cuándo sus aplicaciones están listas para migrar.

Supervise el estado de creación de Replicator

Tras llamar a la `CreateReplicator` API, el replicador pasa por los siguientes estados: `CREATING`, `RUNNING`, o. `FAILED` Espere aproximadamente 30 minutos.

```
aws kafka describe-replicator --replicator-arn <replicator-arn>
```

Supervise la replicación de datos

Supervise las siguientes métricas para realizar un seguimiento del progreso de la replicación de datos:

- **MessageLag**— Supervise hasta que llegue a 0, lo que indica que se han replicado todos los datos.
- **ReplicationLatency**— Realice un seguimiento del tiempo que tardan los registros en replicarse desde el origen al destino.
- **ReplicationFailures**— Debe ser 0. Un valor distinto de cero indica problemas de replicación.

Supervise la sincronización de las compensaciones entre grupos

Cuando `synchroniseConsumerGroupOffsets` está configurado en `true`, MSK Replicator traduce y sincroniza periódicamente las compensaciones de los grupos de consumidores desde el clúster de origen al clúster de destino.

- **ConsumerGroupCount**— Compruebe que coincide con el número esperado de grupos de consumidores que se van a sincronizar.
- **ConsumerGroupOffsetSyncFailure**— Debe ser 0. Si es mayor que 0, compruebe que los grupos de consumidores estén activos, compruebe los permisos de lectura y descripción y asegúrese de que los temas existan en el clúster de destino.
- **OffsetLag (MSK)** y **OffsetLag (Non-MSK)**: compare el retraso de consumo a nivel de partición en ambos clústeres para comprobar que las compensaciones estén sincronizadas.

Determine la preparación para la migración

Sus aplicaciones estarán listas para migrar cuando se cumplan todas las condiciones siguientes:

- **MessageLag** = 0
- **ReplicationFailures** = 0
- **ConsumerGroupOffsetSyncFailure** = 0

Registros de MSK Replicator

MSK Replicator admite el reenvío de registros operativos para ayudarlo a monitorear y solucionar problemas de replicación. La entrega de registros está deshabilitada de forma predeterminada. Al habilitar la entrega de registros, obtendrá una mayor visibilidad de las operaciones de replicación, los problemas de autenticación y los problemas de configuración.

 Note

Cuando la entrega de registros está habilitada, su función de IAM debe tener los permisos adicionales necesarios para escribir en el destino del registro configurado. Para conocer los permisos necesarios, consulte [Habilitar el registro desde AWS los servicios](#).

Destinos de registro compatibles

MSK Replicator admite los siguientes destinos de registro:

- Amazon CloudWatch Logs: vea los registros en tiempo real en Consola de administración de AWS. Admite consultas de información de registros.
- Amazon S3: almacene registros de forma duradera para su retención a largo plazo y análisis de lotes.
- Amazon Data Firehose: transmita los registros a destinos compatibles, como Amazon OpenSearch Service, Splunk o un punto de enlace HTTP personalizado.

Configure la entrega de registros

Configure la entrega de registros incluyendo el `logDelivery` parámetro en su solicitud `CreateReplicator` o en la de la `UpdateReplicator` API. De forma predeterminada, la entrega de registros no está habilitada.

Ejemplo `logDelivery` de configuración:

```
# In CreateReplicator or UpdateReplicator API request:
"logDelivery": {
  "brokerLogs": {
    "cloudWatchLogs": {
      "enabled": true,
      "logGroup": "/aws/msk/replicator/my-replicator"
    },
    "s3": {
      "enabled": true,
      "bucket": "my-msk-logs-bucket",
      "prefix": "msk-replicator/"
    },
    "firehose": { "enabled": false }
  }
}
```

```
}
```

Puede habilitar uno o más destinos simultáneamente en función de sus requisitos de registro.

Contenido del registro

Los registros de MSK Replicator incluyen información sobre los siguientes eventos y condiciones:

1. Fallos de autorización para temas específicos, errores en el tamaño de los registros cuando los mensajes superan los límites configurados y otros errores comunes
2. Compense la actividad de confirmación que confirme el progreso de
3. Creación de temas replicados en el clúster de destino
4. Advertencias y errores de conectividad del clúster

Consideraciones sobre costos

Para obtener más información sobre los precios de los registros vendidos, consulta la pestaña Logs en [Amazon CloudWatch Pricing](#).

Prácticas recomendadas

En esta sección, se describen las prácticas recomendadas y las estrategias de implementación más comunes para utilizar el Replicador Amazon MSK.

Temas

- [Dimensionamiento y particiones](#)
- [Administrar el rendimiento con cuotas de Kafka](#)
- [Periodo de retención](#)
- [La configuración del clúster cambia tras la creación del replicador](#)
- [Consideraciones para aplicaciones multirregionales](#)

Dimensionamiento y particiones

La cantidad de particiones de los clústeres de MSK de origen y destino afecta directamente al rendimiento de la replicación. Tener muy pocas o demasiadas particiones puede afectar al rendimiento.

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

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.

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.

Administrar el rendimiento con 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 depende de la capacidad de lectura del clúster de origen y del rendimiento de los datos que se están replicando.

Puede establecer cuotas de Kafka para el replicador en los clústeres de origen y destino a fin de controlar la capacidad que puede utilizar el replicador de 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.
3. Con las herramientas 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>
```

4. Tras ejecutar el comando, compruebe que la `ReplicatorThroughput` métrica no supere la cuota que ha establecido.

Si reutilizas una función 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 un valor extremadamente bajo, su replicador de MSK `consumer_byte_rate` podría actuar de forma inesperada.

Periodo de retención

Puede establecer el período de retención de registros para los clústeres aprovisionados y sin servidor de MSK. El periodo de retención recomendado es de 7 días. Consulte [La configuración del clúster cambia tras la creación del replicador](#) o [Configuración de clústeres sin servidor de MSK](#).

La configuración del clúster cambia tras la creación del replicador

- Se recomienda no activar ni desactivar el almacenamiento en niveles una vez creado el MSK Replicator. Si el clúster de destino no está agrupado en niveles, MSK no copiará las configuraciones de almacenamiento en niveles, independientemente de si el clúster de origen lo está. 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.
- No cambie las siguientes opciones de configuración del clúster después de crear MSK Replicator, ya que se validan durante la creación:
 - 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.
- En el caso de configuraciones de replicación con nombres de temas idénticos, no modifique los encabezados que crea MSK Replicator (`__msk__`) para evitar el riesgo de una replicación cíclica.

Consideraciones para aplicaciones multirregionales

Al crear aplicaciones Apache Kafka multirregionales con MSK Replicator, tenga en cuenta lo siguiente:

- Consumidores idempotentes: sus consumidores deben poder volver a procesar los mensajes duplicados sin que ello repercuta posteriormente. MSK Replicator replica los datos at-least-once, lo que puede provocar duplicados en el clúster en espera. Al cambiar a la AWS región secundaria, es posible que sus 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.
- Tolerar una pérdida mínima de datos: los productores y los consumidores deben tolerar una pérdida mínima de datos. Dado que MSK Replicator replica los datos de forma asíncrona, cuando la AWS región principal 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.

Solución de problemas de Amazon MSK Replicator

La siguiente información puede ayudarle a solucionar problemas con MSK Replicator. Consulte [Solución de problemas del clúster de Amazon MSK](#) para conocer otras funciones de Amazon MSK. También puede publicar el problema en [AWS re:Post](#).

El estado del replicador pasa de CREATIVO a FALLIDO

Causas comunes del error de creación del MSK Replicator:

1. Compruebe que los grupos de seguridad que proporcionó para el clúster de destino tengan reglas de salida que permitan el tráfico a los grupos de seguridad de su clúster de destino y que los grupos de seguridad de su clúster de destino tengan reglas de entrada que acepten el tráfico de los grupos de seguridad de Replicator.
2. Para la replicación entre regiones, compruebe que el clúster de origen tenga activada la conectividad de varias VPC para el control de acceso de IAM y que la política de clúster esté configurada en el clúster de origen.
3. Compruebe que la función de IAM proporcionada durante la creación tenga los permisos necesarios para leer y escribir en los clústeres de origen y destino, incluidos los permisos para escribir en los temas.
4. Compruebe que su red ACLs no bloquee la conexión entre el MSK Replicator y sus clústeres.
5. Es posible que los clústeres de origen o destino no estén completamente disponibles cuando el MSK Replicator intentó conectarse. Esto puede deberse a una carga excesiva, al uso del disco o al uso de la CPU. 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 parece atascado en el estado DE CREACIÓN

La creación de MSK Replicator puede tardar hasta 30 minutos. Espere 30 minutos y compruebe de nuevo el estado del replicador.

Replicator no replica datos o solo replica datos parciales

1. Compruebe que su replicador no tenga errores de autenticación utilizando la `AuthError` métrica de Amazon CloudWatch. Si esta métrica es superior a 0, compruebe la política de funciones de

- IAM y asegúrese de que no se hayan establecido permisos de denegación para los permisos del clúster.
2. Compruebe que los clústeres de origen y destino no tengan problemas (demasiadas conexiones, disco a plena capacidad o uso elevado de la CPU).
 3. Compruebe que se pueda acceder a los clústeres mediante la `KafkaClusterPingSuccessCount` métrica. Si esta métrica es 0 o no tiene ningún punto de datos, compruebe los permisos de red y de los roles de IAM.
 4. Compruebe que su replicador no tenga errores utilizando la métrica. `ReplicatorFailure` Si es superior a 0, compruebe la función de IAM para ver los permisos de nivel temático.
 5. Compruebe que la expresión regular de la lista de temas permitidos coincide con los nombres de los temas que desea replicar y que los temas no están excluidos de la lista de rechazados.
 6. El replicador puede tardar hasta 30 segundos en detectar y crear temas nuevos. Los mensajes generados antes de la creación del tema en el clúster de destino no se replicarán si la posición inicial es la última (opción predeterminada).

Los desplazamientos de mensajes en el clúster de destino son diferentes a las del clúster de origen

MSK Replicator consume los mensajes del clúster de origen y los envía al clúster de destino, lo que puede generar diferentes compensaciones. Si ha activado la sincronización de compensaciones de grupos de consumidores, MSK Replicator traducirá automáticamente las compensaciones para que, tras la conmutación por error, sus consumidores puedan reanudar el procesamiento casi desde donde lo dejaron.

Replicator no sincroniza las compensaciones de grupos de consumidores

1. Compruebe que la replicación de datos funciona según lo previsto.
2. Compruebe que la expresión regular de la lista de usuarios permitidos coincide con los grupos de consumidores que desea replicar.
3. Compruebe que el Replicador MSK haya creado el tema en el clúster de destino. Si el grupo de consumidores del clúster de origen solo ha consumido los mensajes que no se han replicado, el grupo de consumidores no se replicará en el clúster de destino. Cuando su grupo de consumidores comience a leer los mensajes recién replicados, MSK Replicator replicará automáticamente el grupo de consumidores.

Note

MSK Replicator optimiza la sincronización de las compensaciones de los grupos de consumidores para los consumidores que lean casi al final de la partición temática. Si sus grupos de consumidores están rezagados en el clúster de origen, es posible que vea un retraso mayor en el clúster de destino. 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

1. Compruebe que tiene el número correcto de particiones. La siguiente tabla muestra el número mínimo de particiones recomendado para el rendimiento deseado.

Rendimiento y número mínimo recomendado de particiones

Rendimiento (MB/s)	Se requieren particiones mínimas
50	167
100	334
250	833
500	1666
1 000	3333

2. Compruebe que tiene suficiente capacidad de lectura y escritura en sus clústeres. El Replicador MSK actúa como consumidor del clúster de origen (salida) y como productor del clúster de destino (entrada). Aprovechne la capacidad del clúster para admitir el tráfico de replicación, además de otro tipo de tráfico.
3. La latencia de replicación varía según la distancia entre pares de regiones.
4. Compruebe que su replicador no se esté viendo limitado utilizando la métrica. `ThrottleTime`. Si es superior a 0, ajuste las cuotas de Kafka. Consulte [Administrar el rendimiento con cuotas de Kafka](#).
5. Consulte el [AWS Service Health Dashboard](#) para ver los eventos de servicio de MSK en su región.

Solución de problemas mediante la métrica ReplicatorFailure

La ReplicatorFailure métrica le ayuda a monitorear y detectar problemas de replicación. Un valor distinto de cero normalmente indica un error de replicación provocado por limitaciones en el tamaño de los mensajes, infracciones del intervalo de fechas o problemas con el tamaño de los lotes de registros. Si ha configurado la entrega de registros para su replicador, puede usar los mensajes de registro entregados para identificar el error específico. Para obtener más información, consulte [Registros de MSK Replicator](#). Si la entrega de registros no está configurada, siga los pasos que se indican a continuación para consultar los mensajes de error en el tema de estado del replicador.

Si la ReplicatorFailure métrica indica un valor distinto de cero, siga estos pasos para solucionar el problema:

1. Configure un cliente que pueda conectarse al clúster de MSK de destino y que tenga configuradas las herramientas CLI de Apache Kafka. Consulte [Conexión a un clúster de Amazon MSK aprovisionado](#).
2. ¿Abrir la consola Amazon MSK en <https://console.aws.amazon.com/msk/casa?region=us-east-1#/home/>.

[Obtenga el replicador ARNs de MSK y el clúster de MSK de destino, y obtenga los puntos finales intermediarios del clúster de MSK de destino.](#)

3. Exporte el ARN de MSK Replicator y los puntos finales del broker:

```
export TARGET_CLUSTER_SERVER_STRING=<BootstrapServerString>
export REPLICATOR_ARN=<ReplicatorARN>
export CONSUMER_CONFIG_FILE=<ConsumerConfigFile>
```

4. En su <path-to-your-kafka-installation>/bin directorio, guarde el siguiente script como. **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 --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
    esac
done

```



```

        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

```

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 muestra todos los errores junto con sus mensajes de excepción y las particiones de temas afectadas. Dado que el tema contiene todos los mensajes históricos de error, inicie la investigación con el último mensaje. El siguiente es 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 y soluciones comunes

A continuación, se describen las fallas comunes de MSK Replicator y cómo mitigarlas.

Tamaño del mensaje mayor que max.request.size

Causa: el tamaño del mensaje individual supera los 10 MB (el máximo predeterminado).

Lo siguiente es 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: reduzca el tamaño de los mensajes individuales de su tema. Si no puede, siga las instrucciones para [solicitar un aumento del límite](#).

Tamaño del mensaje mayor que el tamaño máximo de mensaje que el servidor acepta

Causa: el tamaño del mensaje supera el tamaño máximo de mensaje del clúster de destino.

Lo siguiente es 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 del tema o clúster de destino.

Consulte [max.message.bytes](#).

La marca de tiempo está fuera del intervalo

Causa: la marca de tiempo del mensaje está fuera del rango permitido por el clúster de destino.

Lo siguiente es 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 configuración del clúster de `message.timestamp.before.max.ms` destino. Consulte [message.timestamp.before.max.ms](#).

Lote de registros demasiado grande

Causa: el tamaño del lote de registros supera el tamaño de segmento establecido para el tema en el clúster de destino. El Replicador de MSK admite un tamaño máximo de lote de 1 MB.

Lo siguiente es 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: actualice el clúster de destino `segment.bytes` para que sea al menos 1048576 (1 MB). 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.

Solucione los problemas de replicación desde clústeres Kafka autogestionados

MSK Replicator no se puede conectar al clúster Kafka autogestionado

Realice las siguientes comprobaciones si MSK Replicator no puede conectarse a su clúster Kafka autogestionado:

1. Compruebe que su conexión VPN o Direct Connect esté activa y que las tablas de enrutamiento sean correctas.
2. Compruebe que los grupos de seguridad permitan el tráfico entrante desde las subredes de MSK Replicator en el SASL_SSL puerto (normalmente 9096).

3. Compruebe la resolución de DNS de la VPC a los nombres de host del agente de clústeres autogestionado.
4. Compruebe la `KafkaClusterPingSuccessCount` métrica en Amazon CloudWatch : un valor de 0 indica un fallo de conectividad.

Fallos de autenticación de SASL/SCRAM

Si la `AuthError` métrica es distinta de cero o los registros del Replicador muestran errores: SASL/SCRAM

1. Compruebe que las credenciales almacenadas en AWS Secrets Manager coincidan con las credenciales de usuario de SCRAM del clúster autogestionado.
2. Compruebe que el usuario de SCRAM tiene los permisos de ACL necesarios (leer, describir los temas, leer y describir los grupos de consumidores y describir los temas).
3. Compruebe la `AuthError` métrica para confirmar los errores de autenticación e identificar si el clúster de origen o de destino se ve afectado mediante la `ClusterAlias` dimensión.

Problemas con los certificados SSL

Si el replicador no puede establecer una conexión segura con el clúster autogestionado:

1. Compruebe que el `certificate` valor de AWS Secrets Manager incluya toda la cadena de certificados de CA en formato PEM.
2. Compruebe que el agente de escucha SSL esté configurado en todos los agentes de clústeres autogestionados.
3. Compruebe que el certificado no haya caducado y que lo haya emitido una CA de confianza.

El grupo de consumidores compensa los errores de sincronización en los clústeres autogestionados

Si las compensaciones de los grupos de consumidores no se sincronizan correctamente:

1. Compruebe la `ConsumerGroupOffsetSyncFailure` métrica: debe ser 0.
2. Compruebe que los grupos de consumidores consumen activamente en el clúster de origen (es posible que los grupos de consumidores inactivos no estén sincronizados).

3. Para la replicación bidireccional, compruebe que `synchroniseConsumerGroupOffsets` esté configurada `true` en ambos replicadores.

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](#)
- [AWS Lambda integración con Amazon MSK](#)
- [Acceda a Amazon EventBridge Pipes a través de la consola Amazon MSK](#)
- [Uso de Kafka Streams con agentes Express de MSK y Amazon MSK sin servidor](#)
- [Esquemas 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 Introduce Managed Data Delivery from Apache Kafka to Your Data Lake](#)
- Laboratorio: [Delivery to Amazon S3 using Amazon Data Firehose](#)

AWS Lambda integración con Amazon MSK

La integración de Lambda conecta el clúster de Amazon MSK con la función de Lambda seleccionada mediante una asignación de orígenes de eventos (ESM) que sondea constantemente los mensajes del tema mediante un recurso denominado sondeo de eventos. El ESM evalúa la acumulación de mensajes (utilizando la [OffsetLag métrica](#)) para todas las particiones del tema y escala automáticamente los Event Pollers para procesar los mensajes de manera eficiente.

Para obtener más información, consulte [Uso de Lambda con Amazon MSK](#) en la AWS Lambda Guía para desarrolladores.

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 agentes Express de MSK y Amazon MSK sin servidor

Kafka Streams admite transformaciones sin estado y con estado. Las transformaciones con estado, como el recuento, la agregación o la unión, utilizan operadores que almacenan su estado en temas internos de Kafka. Además, algunas transformaciones sin estado, como `groupBy` o la redistribución de particiones, almacenan sus resultados en temas internos de Kafka. De forma predeterminada, Kafka Streams asigna nombres a 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 de forma rígida la configuración `segment.bytes` y la establece en 50 MB. [MSK aprovisionado con agentes Express](#) y MSK sin servidor protegen algunas [configuraciones de temas](#), incluido el tamaño del segmento, durante la creación del tema. Por lo tanto, una aplicación de Kafka Streams con transformaciones con estado no puede crear los temas internos cuando se utilizan agentes Express de MSK o MSK sin servidor.

Para ejecutar este tipo de aplicaciones de Kafka Streams en agentes Express de MSK o en MSK sin servidor, debe crear los temas internos por su cuenta. Para ello, primero identifique y asigne un nombre a los operadores de Kafka Streams, que requieren temas. A continuación, cree los temas internos de Kafka correspondientes.

Note

- Se considera una práctica recomendada asignar nombres manualmente a los operadores en Kafka Streams, especialmente a aquellos que dependen de temas internos. Para

obtener información sobre la asignación de nombres a operadores, consulte [Asignación de nombres a operadores en una aplicación de DSL de Kafka Streams](#) en la documentación de Kafka Streams.

- El nombre del tema interno de una transformación con estado depende del `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 Express de MSK o MSK sin servidor](#)

Creación de una aplicación de Kafka Streams con agentes Express de MSK o MSK sin servidor

Si la aplicación de Kafka Streams tiene su `application.id` establecido en `msk-streams-processing`, puede crear una aplicación de Kafka Streams con agentes Express de MSK o MSK sin servidor. Para ello, utilice el operador `count()`, que requiere un tema interno con ese nombre. Por ejemplo, `msk-streams-processing-count-store`.

Para crear una aplicación de Kafka Streams, haga lo siguiente:

Temas

- [Cómo identificar operadores y asignarles un nombre](#)
- [Creación de los temas internos](#)
- [\(Opcional\) Comprobación del nombre del tema](#)
- [Ejemplos de asignación de nombres a operadores](#)

Cómo identificar operadores y asignarles un nombre

1. Identifique los procesadores con estado mediante [Transformaciones con estado](#) en la documentación de Kafka Streams.

Algunos ejemplos de procesadores con estado incluyen `count`, `aggregate` o `join`.

2. Identifique los procesadores que crean temas para la redistribución de particiones.

El siguiente ejemplo contiene una operación `count()`, que requiere estado.

```
var stream =  
    paragraphStream  
        .groupByKey()  
        .count()  
        .toStream();
```

3. Para asignar un nombre al tema, agregue un nombre para cada procesador con estado. Según el tipo de procesador, la asignación de nombres se realiza mediante una clase de asignación de nombres diferente. Por ejemplo, la operación `count()` es una operación de agregación. Por lo tanto, requiere la clase `Materialized`.

Para obtener información sobre las clases de asignación de nombres para las operaciones con estado, consulte [Conclusión](#) en la documentación de Kafka Streams.

En el siguiente ejemplo, se establece el nombre del operador `count()` como `count-store` mediante la clase `Materialized`.

```
var stream =  
    paragraphStream  
        .groupByKey()  
        .count(Materialized.<String, Long, KeyValueStore<Bytes, byte[]>>as("count-  
store") // descriptive name for the store  
            .withKeySerde(Serdes.String())  
            .withValueSerde(Serdes.Long()))  
        .toStream();
```

Creación de los temas internos

Kafka Streams antepone el prefijo `application.id` a los nombres de los temas internos, donde `application.id` es definido por el usuario. Por ejemplo, `application.id-internal_topic_name`. Los temas internos son temas normales de Kafka, y puede crear los temas con la información disponible en [Creación de un tema de Apache Kafka](#) o `AdminClient` de la API de Kafka.

Según el caso de uso, puede utilizar las políticas predeterminadas de limpieza y retención de Kafka Streams o personalizar sus valores. Estas políticas se definen en `cleanup.policy` y `retention.ms`.

En el siguiente ejemplo, se crean los temas con la API de AdminClient y se establece el `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 que los temas se han creado en el clúster, su aplicación de Kafka Streams puede utilizar el tema `msk-streams-processing-count-store` para la operación `count()`.

(Opcional) Comprobación del nombre del tema

Puede utilizar el descriptor de topología para describir la topología del flujo y ver los nombres de los temas internos. En el siguiente ejemplo, se muestra cómo ejecutar el descriptor de topología.

```
final StreamsBuilder builder = new StreamsBuilder();
Topology topology = builder.build();
System.out.println(topology.describe());
```

La siguiente salida 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 usar el descriptor de topología, consulte [Asignación de nombres a operadores en una aplicación de DSL de Kafka Streams](#) en la documentación de Kafka Streams.

Ejemplos de asignación de nombres a operadores

Esta sección proporciona algunos ejemplos de asignación de nombres a operadores.

Ejemplo de operador de nomenclatura para groupByKey ()

```
groupByKey() -> groupByKey(Grouped.as("kafka-stream-groupby"))
```

Ejemplo de asignación de nombre a un operador para normal count()

```
normal count() -> .count(Materialized.<String, Long, KeyValueStore<Bytes,  
    byte[]>>as("kafka-streams-window") // descriptive name for the store  
    .withKeySerde(Serdes.String())  
    .withValueSerde(Serdes.Long()))
```

Ejemplo de asignación de nombre a un operador 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 asignación de nombre a un operador para windowed suppressed()

```
windowed suppressed() ->  
Suppressed<Windowed> suppressed = Suppressed  
    .untilWindowCloses(Suppressed.BufferConfig.unbounded())  
    .withName("kafka-suppressed");  
    .suppress(suppressed)
```

Esquemas de incrustación vectorial en tiempo real

Amazon MSK (Managed Streaming for Apache 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 esquema de MSF simplifica la incorporación de los datos más recientes de las canalizaciones de streaming de Amazon MSK en los 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:
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
- Canadá (centro): 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](#)
- [Notas antes de habilitar esquemas de incrustación vectorial en tiempo real](#)

- [Implementación de un esquema de vectorización de datos en 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 disponibles para una aplicación MSF estándar y para Amazon Bedrock pueden supervisar la [aplicación](#) y las [métricas de Bedrock](#).

Existen dos métricas adicionales para supervisar 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 el OpenSearch servicio, puedes usar las siguientes métricas:

- 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.

Notas antes de habilitar esquemas de incrustación vectorial en tiempo real

La aplicación de Managed Service para Apache Flink solo admite texto no estructurado o datos en formato JSON en el flujo de entrada.

Se admiten dos modos de procesamiento de entrada:

- Cuando los datos de entrada son texto no estructurado, se incrusta el mensaje de texto completo. 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 varias 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 conjuntamente 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 exclusivamente todos los modelos de incrustación de texto proporcionados por Bedrock.

Persisten en un 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 use OpenSearch sin servidor para crear un índice vectorial, utilice siempre el nombre del campo vectorial `embedded_data`.

Al igual que con los esquemas de MSF, se espera que administre la infraestructura necesaria para ejecutar el código asociado con el esquema 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.

Implementación de un esquema de vectorización de datos en streaming

Este tema describe cómo implementar un esquema de vectorización de datos en streaming.

Implementación de un esquema de vectorización de datos en streaming

1. Asegúrese de que los siguientes recursos estén configurados correctamente:
 - Un clúster de Amazon MSK aprovisionado o sin servidor con uno o más temas que contengan datos.
2. Configuración de Bedrock: [Acceso al modelo de Bedrock deseado](#). Modelos de Bedrock compatibles actualmente:
 - 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 OpenSearch de servicios debe tener al menos un índice.
- Si planea usar una colección OpenSearch sin servidor, asegúrese de crear una colección de búsqueda vectorial. Para obtener información detallada sobre cómo configurar un índice vectorial, consulte [Requisitos previos para usar un almacén vectorial propio para 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 piensa utilizar 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 [habilitar el control de acceso detallado](#).
 - Si lo desea, puede habilitar el acceso al OpenSearch panel de control para ver los resultados. Consulte [habilitar el control de acceso detallado](#).
4. Inicie sesión con un rol que permita CreateStack los permisos [aws](#):
 5. Vaya al panel de la consola de MSF y seleccione Crear aplicación de streaming.
 6. En Elegir un método para configurar la aplicación de procesamiento de flujos, seleccione Usar un esquema.
 7. Seleccione Esquema de aplicación de IA en tiempo real en el menú desplegable de esquemas.
 8. Proporcione las configuraciones deseadas. Consulte [Creación de configuraciones de la página](#).
 9. Seleccione Deploy Blueprint para iniciar una CloudFormation implementación.
 10. Una vez completada la CloudFormation implementación, vaya a la aplicación Flink implementada. Revise las propiedades de tiempo de ejecución de la aplicación.
 11. Puede elegir ejecutar las change/add propiedades de su aplicación. Para obtener más información, consulte [Configuración de propiedades de tiempo de ejecución](#) para conocer los detalles necesarios para configurar estas propiedades.

Note

Nota:

Si utilizas OpenSearch aprovisionadas, asegúrate de activar 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, asegúrese de que la aplicación de MSF tenga acceso 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 que esté conforme con todas las configuraciones, seleccione Run. La aplicación comenzará a ejecutarse.
13. Envíe mensajes al clúster de MSK.
14. Navegue hasta el clúster de OpenSearch y vaya al panel de control. OpenSearch
15. En el panel, seleccione Descubrir en el menú de la izquierda. Debería ver los documentos almacenados junto con sus incrustaciones vectoriales.
16. Consulte [Trabajo con colecciones de búsqueda vectorial](#) para conocer cómo puede utilizar los vectores almacenados en el índice.

Creación de configuraciones de la página

Este tema describe las configuraciones de página que se deben consultar al especificar las configuraciones de esquemas de aplicaciones de IA en tiempo real.

Nombre de la aplicación

Campo existente en MSF: asigne cualquier nombre a la aplicación.

Clúster de MSK

Seleccione el clúster de MSK que creó durante la configuración desde la lista desplegable.

Temas

Agregue el nombre del tema o de los temas que creó durante la configuración.

Tipo de datos del flujo de entrada

Seleccione Cadena si va a proporcionar entrada de tipo cadena al flujo de MSK.

Seleccione JSON si la entrada en el flujo de MSK está en formato JSON. En claves JSON incrustadas, escriba los nombres de los campos del JSON de entrada cuyo valor desea 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 [Agregar o eliminar acceso a los modelos base de Amazon Bedrock](#).

OpenSearch clúster

Seleccione el clúster que creó en la lista 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 está Region-specific dentro de sus límites. Cuenta de AWS

Temas

- [Solicitud de un aumento de cuota en Amazon MSK](#)
- [Cuota estándar de agentes de Amazon MSK](#)
- [Cuota de agentes Express de Amazon MSK](#)
- [Cuotas del Replicador MSK](#)
- [Cuota de MSK sin servidor](#)
- [Cuota de MSK Connect](#)

Solicitud de 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](#).

El soporte puede aprobar, denegar o aprobar parcialmente las 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 **msk** y, a continuación, elija Amazon Managed Streaming para Apache Kafka (MSK).
5. En Cuotas de servicio elija el nombre de la cuota para la que desea solicitar un aumento. Por ejemplo, **Number of brokers per account**.
6. Elija Solicitud de aumento a nivel de cuenta.
7. En Aumentar valor de cuota, introduzca un nuevo valor de cuota.

8. Seleccione Solicitar.
9. (Opcional) Para ver cualquier solicitud pendiente o resuelta recientemente en la consola, elija Panel 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. Después de que el estado cambie a Cuota solicitada, verá el número de caso con el soporte. Elija el número de caso para abrir el ticket para su solicitud.

Para obtener más información, incluido cómo usar los AWS CLI SDK para solicitar un aumento de cuota, consulte [Solicitar un aumento de cuota](#) en la Guía del usuario de Service Quotas.

Cuota estándar de agentes de Amazon MSK

La siguiente tabla describe las cuotas para agentes estándar.

Dimensión	Cuota	Notas
Agentes por cuenta	90	Para solicitar una cuota más alta, vaya a la consola de Service Quotas .
Agentes por clúster	30 para ZooKeeper-based clústeres, 60 para KRaft-based clústeres.	Para solicitar una cuota más alta, vaya a la consola de Service Quotas .
Almacenamiento mínimo por agente	1 GiB	
Almacenamiento máximo por agente	16384 GiB	
Máximo de conexiones TCP por agente (control de acceso de IAM)	3 000	Para aumentar este límite, puede ajustar <code>listener.name.client_iam.max.connections</code> o la propiedad de configuración <code>listener.name.client_iam_public.max.</code>

Dimensión	Cuota	Notas
		connections mediante la API AlterConfig de Kafka o la herramienta kafka-configs.sh. Es importante tener en cuenta que aumentar cualquiera de las propiedades a un valor alto puede provocar la falta de disponibilidad.
Tasa máxima de conexiones TCP por agente (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 desea que un cliente reintente las conexiones después de 1 segundo, establezca reconnect.backoff.ms en 1000. Para obtener más información, consulte reconnect.backoff.ms en la documentación de Apache Kafka.
Máximo de conexiones TCP por agente (sin IAM)	N/A	MSK no aplica límites de conexión para la autenticación sin IAM. Debe supervisar otras métricas, como el uso de CPU y memoria, para asegurarse de no sobrecargar el clúster debido a un número excesivo 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 Dimensione correctamente el clúster: número de particiones por agente Standard.
Revisiones de configuración por cuenta	50	

Cuota de agentes Express de Amazon MSK

La siguiente tabla describe las cuotas para agentes Express.

Dimensión	Cuota	Notas
Agentes por cuenta	90	Para solicitar una cuota más alta, vaya a la consola de Service Quotas .
Agentes por clúster	30 para ZooKeeper-based clústeres, 60 para clústeres KRaft-based	Para solicitar una cuota más alta, vaya a la consola de Service Quotas .
Almacenamiento máximo	Sin límite	

Dimensión	Cuota	Notas
Máximo de conexiones TCP por agente (control de acceso de 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: <code>listener.name.client_iam.max.connections</code> <code>listener.name.client_iam_public.max.connections</code> Establecer estos valores en un nivel alto puede provocar la indisponibilidad del clúster.
Tasa máxima de conexiones TCP por agente (IAM)	100 por segundo	Para gestionar los reintentos en caso de conexiones fallidas, puede configurar el parámetro de configuración <code>reconnect.backoff.ms</code> en el lado del cliente. Por ejemplo, si desea que un cliente reintente las conexiones después de 1 segundo, establezca <code>reconnect.backoff.ms</code> en 1000. Para obtener más información, consulte reconnect.backoff.ms en la documentación de Apache Kafka.

Dimensión	Cuota	Notas
Máximo de conexiones TCP por agente (sin IAM)	N/A	MSK no aplica límites de conexión para la autenticación sin IAM. No obstante, debe supervisar otras métricas, como el uso de CPU y memoria, para asegurarse de no sobrecargar el clúster debido a un número excesivo de conexiones.
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 Dimensione correctamente el clúster: número de particiones por agente Express .
Revisiones de configuración por cuenta	50	
Ingreso máximo por agente	Recomendado: 15,6 - 500,0 MBps	Según el tamaño de la instancia
Salida máxima por agente	Recomendado: 31,2 - 1000,0 MBps	Según el tamaño de la instancia
Rendimiento por partición	Máximo 15 MB/s	

Temas

- [Límites de limitación controlada del rendimiento de agentes Express según el tamaño del agente](#)
- [Cuota de particiones de agentes Express](#)

Límites de limitación controlada del rendimiento de agentes Express según el tamaño del agente

En la siguiente tabla, se enumeran los límites de limitación controlada del rendimiento recomendados y máximos relacionados con el ingreso y la salida para distintos tamaños de agente. En esta tabla, el rendimiento recomendado se representa como Rendimiento sostenido, que es el umbral hasta el cual las aplicaciones no experimentarán degradación del rendimiento. Si opera más allá de estos límites en cualquiera de las dos dimensiones, puede obtener mayor rendimiento, pero también podría experimentar degradación del rendimiento. La cuota máxima es el umbral en el que el clúster limitará el read/write tráfico. Las aplicaciones no podrán operar más allá de este umbral.

Tamaño de instancia	Rendimiento sostenido (MBps) para ingreso	Cuota máxima (MBps) para ingreso	Rendimiento sostenido (MBps) para salida	Cuota máxima (MBps) para salida
express.m 7g.large	15.6	23.4	31,2	58,5
express.m 7g.xlarge	31,2	46,8	62,5	117
express.m 7g.2xlarge	62,5	93,7	125	234.2
express.m 7g.4xlarge	124,9	187,5	249,8	468,7
express.m 7g.8xlarge	250	375	500	937,5
express.m 7g.12xlarge	375	562,5	750	1406.2

Tamaño de instancia	Rendimiento sostenido (MBps) para ingreso	Cuota máxima (MBps) para ingreso	Rendimiento sostenido (MBps) para salida	Cuota máxima (MBps) para salida
express.m7g.16xlarge	500	750	1 000	1875

Cuota de particiones de agentes Express

La siguiente tabla muestra el número recomendado de particiones (incluidas las réplicas líder y seguidora) para cada agente Express. No puede superar el número máximo de particiones indicado en la siguiente tabla para cada agente Express.

Para obtener información sobre las prácticas recomendadas que se deben tener en cuenta al asignar particiones a los agentes Express, consulte [Dimensione correctamente el clúster: número de particiones por agente Express](#).

Tamaño del corredor	Número recomendado de particiones (incluidas las réplicas de seguidor y líder) por agente	Número máximo de particiones por agente
express.m7g.large	1 000	1500
express.m7g.xlarge	1 000	2000
express.m7g.2xlarge	2500	4000
express.m7g.4xlarge	6000	8000
express.m7g.8xlarge	12000	16 000
express.m7g.12xlarge	16 000	24000
express.m7g.16xlarge	20000	32000

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. Para solicitar soporte para más de 750 temas por replicador, póngase en contacto con AWS Support. Puede controlar la cantidad de temas que se replican mediante la `TopicCount` métrica.
- Un rendimiento de entrada máximo de 1 GB por segundo por MSK Replicator. Para solicitar una cuota mayor, ponte en contacto con AWS Support.
- Tamaño de registro de MSK Replicator: un tamaño de registro máximo de 10 MB para la replicación entre regiones y 20 MB para la replicación en la misma región ().
`message.max.bytes` Para solicitar una cuota mayor, ponte en contacto con AWS Support.

Cuota de MSK sin servidor

Las cuotas especificadas en la siguiente tabla se aplican por clúster, salvo 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

Dimensión	Cuota	Resultado de infracción de cuota
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 a las API de administración de temas	2 por segundo	Ralentización con la duración del acelerador en respuesta
Número máximo de bytes de recuperación por solicitud	55 MB	La solicitud falla con: INVALID_REQUEST ErrorCode
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

Dimensión	Cuota	Resultado de infracción de cuota
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 VPC de cliente por clúster sin servidor	5	
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
Actualización de la política de funciones vinculadas al servicio para la conectividad IPv6	Amazon MSK actualizó el <code>KafkaServiceRolePolicy</code> para añadir soporte de IPv6 conectividad. La política ahora incluye <code>ec2:AssignIpv6Addresses</code> , <code>ec2:UnassignIpv6Addresses</code> , y <code>ec2:ModifyNetworkInterfaceAttribute</code> permisos para habilitar la IPv6 conectividad de los clústeres de MSK. Para obtener más información, consulte KafkaServiceRolePolicy .	17-11-2020
Compatibilidad con Apache Kafka versión 4.1.x.	Amazon MSK ahora admite Apache Kafka versión 4.1.x. Para obtener más información, consulte Versiones de Apache Kafka compatibles .	2025-10-15
Mejora del tema de migración de cargas de trabajo de Kafka a un clúster de Amazon MSK	Se mejoró el tema sobre la migración de cargas de trabajo a Apache Kafka con	12/10/2025

Cambio	Descripción	Fecha
	una guía mejorada sobre el uso de MSK Replicator o Apache 2.0 para la migración . MirrorMaker Además, mejoró la organización del contenido e incluyó enlaces relevantes a recursos que proporcionan step-by-step orientación sobre la migración. Para obtener más información, consulte Migración de cargas de trabajo de Kafka a un clúster de Amazon MSK .	
Lanzamiento de mTLS en las regiones de China	Amazon MSK ahora admite mTLS para autenticar clientes en las regiones de China (Pekín) y China (Ningxia) Para obtener más información, consulte Autenticación y autorización para Apache Kafka. APIs	22-09-2020

Cambio	Descripción	Fecha
Mejora del tema sobre versiones de Apache Kafka	- Se ha mejorado la documentación de la versión 3.9 de Apache Kafka, con detalles de soporte ampliados y aclaración de que es la última versión compatible con los sistemas de administración de metadatos y los sistemas de administración de metadatos ZooKeeper . KRaft - Se mejoró el procedimiento de actualización de versión de Kafka con orientación más detallada. Para obtener más información, consulte Versiones compatibles de Apache Kafka y Actualizar la versión de Apache Kafka.	2025-09-19

Cambio	Descripción	Fecha
StorageUsed métrica para corredores de Express	Amazon MSK ahora incluye una nueva métrica de nivel PREDETERMINADO (StorageUsed) para agentes Express que proporciona visibilidad en el clúster sobre el consumo total de almacenamiento, excluidas las réplicas. Para obtener más información, consulte Supervisión de nivel PREDETERMINADO para agentes Express .	24-07-2020
Número elevado de particiones para agentes Express de Amazon MSK	Amazon MSK lanzó compatibilidad con un alto número de particiones para agentes Express. Para obtener más información, consulte Cuota de particiones de agentes Express .	2025-07-21
Nuevas métricas de Amazon MSK Connect	MSK Connect agregó dos métricas nuevas (SinkConsumerByteRate y SourceProducerByteRate) para medir las tasas de rendimiento de datos de los conectores. Para obtener más información, consulte Supervisión de MSK Connect .	2025-06-30

Cambio	Descripción	Fecha
Revisión de los temas de introducción de MSK aprovisionado	Se reestructuraron por completo los temas de introducción de MSK aprovisionado para mejorar la experiencia del usuario, el flujo del contenido y la legibilidad. Los temas revisados también incluyen documentación detallada de todas las opciones de la consola, incluidos los modos de almacenamiento, los métodos de autenticación y los niveles de supervisión, con una orientación clara para la toma de decisiones. Para obtener más información, consulte Introducción a Amazon MSK.	2025-06-28
Compatibilidad con agentes Express en Apache Kafka versión 3.8.x	Amazon MSK ahora admite agentes Express en Apache Kafka versión 3.8.x. Para obtener más información, consulte Agentes Express de Amazon MSK.	2025-06-05

Cambio	Descripción	Fecha
Nueva información de solución de problemas del Replicador de Amazon MSK	La Guía para desarrolladores de Amazon Managed Streaming para Apache Kafka ahora incluye documentación integral sobre la solución de problemas del Replicador de MSK, con un script de diagnóstico y procedimientos detallados de mitigación de errores. Para obtener más información, consulte Solución de problemas de errores de MSK Replicator mediante métricas. ReplicatorFailure	2025-05-09
Renovaciones de certificados sin interrupciones	Amazon MSK lanzó renovaciones de certificados sin interrupciones para agentes Express con el fin de eliminar el tiempo de inactividad de mantenimiento durante las actualizaciones obligatorias de certificados de 13 meses. Para obtener más información, consulte Cifrado de Amazon MSK .	2025-05-05
Compatibilidad con Apache Kafka versión 4.0.x	Amazon MSK ahora admite Apache Kafka versión 4.0.x. Para obtener más información, consulte Versiones de Apache Kafka compatibles .	2025-05-02

Cambio	Descripción	Fecha
Lanzamiento de Amazon MSK Connect en las regiones de China	MSK Connect ahora está disponible en todas las regiones de China: China (Pekín) y China (Ningxia).	2025-04-10
Compatibilidad con Apache Kafka versión 3.9.x	Amazon MSK ahora admite Apache Kafka versión 3.9.x. Para obtener más información, consulte las versiones compatibles de Apache Kafka .	2025-04-21
Conector de fregadero Amazon EventBridge Kafka para MSK Connect	La guía para desarrolladores de Amazon Managed Streaming for Apache Kafka incluye ahora un tema completo que describe cómo utilizar el conector tipo colector EventBridge Kafka con MSK Connect. Para obtener más información, consulte Configurar el conector colector EventBridge Kafka para MSK Connect .	28-03-2021
Actualización de cuotas de los agentes Express	La Guía para desarrolladores de Amazon Managed Streaming para Apache Kafka ahora incluye información sobre los límites de rendimiento de ingreso y salida de datos para los agentes Express. Para obtener más información, consulte Cuota de agentes Express de Amazon MSK .	2025-03-06

Cambio	Descripción	Fecha
Compatibilidad con Apache Kafka versión 3.8.x	Amazon MSK ahora admite Apache Kafka versión 3.8.x. Para obtener más información, consulte Versiones de Apache Kafka compatibles .	2025-02-20
Revisión de la fecha de finalización de soporte de Amazon MSK versión 3.4.0	La fecha de finalización de soporte revisada para Apache Kafka versión 3.4.0 es el 4 de agosto de 2025. Para obtener más información, consulte Versiones de Apache Kafka compatibles .	2025-02-18
UpdateConnector Lanzamiento de la API para modificar las configuraciones de los conectores MSK Connect existentes	Amazon MSK ahora incluye la UpdateConnector API para modificar las configuraciones de los conectores MSK Connect existentes, lo que elimina la necesidad de crear nuevos conectores. Además, se agregó documentación sobre las operaciones de actualización de los conectores DescribeConnectorOperations ListConnectorOperations APIs para realizar un seguimiento de las mismas y mantener registros de auditoría históricos de los cambios de configuración.	12-01-2021

Cambio	Descripción	Fecha
AWS PrivateLink documenta ción de puntos finales de VPC de interfaz	La guía para desarroll adores de Amazon Managed Streaming for Apache Kafka ahora incluye documentación sobre los puntos de enlace de la interfaz de la AWS PrivateLi nk VPC. Para obtener más información, consulte Uso de Amazon MSK APIs con puntos de enlace de VPC de interfaz .	18 de diciembre de 2022
Lanzamiento de MSK Connect versión 3.7.x.	MSK Connect ahora admite la versión 3.7.x. Para obtener más información, consulte Comprensión de MSK Connect .	2024-12-18
Se agregó la característica de agentes Express. Se reorganiz aron los temas de la Guía para desarrolladores.	MSK admite agentes Standard y los nuevos agentes 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

Cambio	Descripción	Fecha
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
<code>WriteDataIdempotently</code> añadido a <code>AWSMSKReplicator ExecutionRole</code>	<code>WriteDataIdempotently</code> se ha agregado un permiso a la <code>AWSMSKReplicator ExecutionRole</code> política para admitir la replicación de datos entre clústeres de MSK.	16-05-2022
Se lanzaron los agentes de M7g de Graviton en Brasil y Bahrein.	Amazon MSK ahora admite la disponibilidad de corredores M7g en las regiones de Sudamérica (sa-east-1, São Paulo) y Oriente Medio (me-south-1, Baréin) mediante procesadores Graviton (AWS procesadores personalizados basados en ARM creados por Amazon Web Services).	7 de febrero de 2024

Cambio	Descripción	Fecha
Lanzamiento de los agentes de M7g de Graviton en la región de China	Amazon MSK ahora admite la disponibilidad en la región de China de los corredores de M7g que utilizan procesadores AWS Graviton (procesadores personalizados basados en ARM creados por Amazon Web Services).	11-01-2022
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
Nueva política de roles de ejecución de servicios para la compatibilidad con el Replicador Amazon MSK.	Amazon MSK agregó una nueva política de <code>AWSMSKReplicatorExecutionRole</code> 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 corredores M7g que utilizan procesadores AWS Graviton (procesadores personalizados basados en ARM creados por Amazon Web Services).	27-11-2020

Cambio	Descripción	Fecha
Replicador Amazon MSK	El Replicador Amazon MSK es una nueva característica que puede utilizar para replicar datos entre clústeres de Amazon MSK. Amazon MSK Replicator incluye una actualización de la política de Amazon MSKFull Access. Para obtener más información, consulte Políticas administradas de AWS : AmazonMSK FullAccess .	28 de septiembre de 2021
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

Cambio	Descripción	Fecha
Actualizaciones de funciones vinculadas al servicio para permitir la conectividad privada de varias VPC	Amazon MSK ahora incluye actualizaciones de funciones <code>AWSService RoleForKafka</code> vinculadas a servicios para gestionar las interfaces de red y los puntos de enlace de la VPC de su cuenta, lo que permite que los agentes de clústeres sean accesibles para los clientes de su VPC. Amazon MSK usa permisos para <code>DescribeVpcEndpoints</code> , <code>ModifyVpcEndpoint</code> y <code>DeleteVpcEndpoints</code> . Para obtener más información, consulte Service-linked funciones para Amazon MSK .	08-03-2021
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
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

Cambio	Descripción	Fecha
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
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

Cambio	Descripción	Fecha
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 .	2021-01-19
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 existentes 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 Supervisión de un clúster de Amazon MSK aprovisionado .	2020-11-23
Soporte de Cruise Control	Amazon MSK ahora LinkedIn es compatible con el Cruise Control. Para obtener más información, consulte Utilice LinkedIn el control de crucero para Apache Kafka con Amazon MSK .	17-11-2020

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 clientes y puntos de enlace. ZooKeeper Para obtener más información, consulte Versiones compatibles de Apache Kafka .	30 de septiembre de 2020
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 SASL/SCRAM authentication .	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 los corredores a CloudWatch Logs, Amazon S3 y Amazon Data Firehose. Firehose, a su vez, puede entregar estos troncos a los destinos a los que apoya, 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 características se incluyen la asistencia para el etiquetado, autenticación, cifrado de TLS, configuraciones 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.