aws

Guia do Desenvolvedor

Amazon Managed Streaming for Apache Kafka

Amazon Managed Streaming for Apache Kafka: Guia do Desenvolvedor

Table of Contents

Boas-vindas ao Guia do desenvolvedor do Amazon MSK

Bem-vindo ao Guia do desenvolvedor do Amazon Managed Streaming for Apache Kafka. Os tópicos a seguir podem ajudar você a começar a usar este guia com base no que você estiver tentando fazer.

- Crie um cluster provisionado pelo MSK seguindo o tutorial. Conceitos básicos sobre como usar o Amazon MSK
- Mergulhe mais na funcionalidade do MSK Provisioned in. O que é MSK Provisioned?
- Execute o Apache Kafka sem precisar gerenciar e escalar a capacidade do cluster com o MSK Serverless.
- Use o MSK Connect para transmitir dados de e para seu cluster Apache Kafka.
- Use o MSK Replicator para replicar dados de forma confiável em clusters provisionados do MSK em diferentes ou iguais. Regiões da AWS

Para os destaques, detalhes do produto e preços, consulte a página de serviços do Amazon MSK.

O que é o Amazon MSK?

O Amazon Managed Streaming for Apache Kafka (Amazon MSK) é um serviço totalmente gerenciado que o habilita a criar e executar aplicações que usam o Apache Kafka para processar dados de transmissões. O Amazon MSK fornece as operações do ambiente de gerenciamento, como as operações para criar, atualizar e excluir clusters. Ele permite usar operações do plano de dados do Apache Kafka, como aqueles para produzir e consumir dados. Ele executa versões de código aberto do Apache Kafka. Isso significa que aplicativos, ferramentas e plug-ins existentes de parceiros e da comunidade Apache Kafka são compatíveis sem a necessidade de fazer alterações no código do aplicativo. É possível usar o Amazon MSK para criar clusters com qualquer uma das versões do Apache Kafka listadas em the section called “Versões compatíveis do Apache Kafka”.

Estes componentes descrevem a arquitetura do Amazon MSK:

- Nós de intermediário — Ao criar um cluster do Amazon MSK, você especifica quantos nós do agente deseja que o Amazon MSK crie em cada zona de disponibilidade. O mínimo é de um agente por zona de disponibilidade. Cada zona de disponibilidade tem sua própria sub-rede de nuvem privada virtual (VPC).

O Amazon MSK Provisioned oferece dois tipos de corretores: e. Corretores Amazon MSK Standard Corretores Amazon MSK Express No MSK Serverless, o MSK gerencia os nós do broker usados para lidar com seu tráfego e você só provisiona os recursos do servidor Kafka em nível de cluster.

- ZooKeeper nós — O Amazon MSK também cria os ZooKeeper nós Apache para você. O Apache ZooKeeper é um servidor de código aberto que permite uma coordenação distribuída altamente confiável.
- KRaft controladores — A comunidade Apache Kafka desenvolvida KRaft para substituir o Apache no gerenciamento de metadados nos clusters do Apache ZooKeeper Kafka. No KRaft modo, os metadados do cluster são propagados dentro de um grupo de controladores Kafka, que fazem parte do cluster Kafka, em vez de entre nós. ZooKeeper KRaftos controladores estão incluídos sem custo adicional para você e não exigem configuração ou gerenciamento adicionais de sua parte.
- Produtores, consumidores e criadores de tópicos: o Amazon MSK permite que você use operações do plano de dados do Apache Kafka para criar tópicos, além de produzir e consumir dados.
- Operações de cluster Você pode usar o AWS Management Console, o AWS Command Line Interface (AWS CLI) ou o APIs no SDK para realizar operações no plano de controle. Por exemplo, você pode criar ou excluir um cluster do Amazon MSK, listar todos os clusters em uma conta, visualizar as propriedades de um cluster e atualizar o número e o tipo de agentes em um cluster.

O Amazon MSK detecta e se recupera automaticamente dos cenários de falha mais comuns para clusters, permitindo que as aplicações produtoras e consumidoras possam continuar suas operações de gravação e leitura com o menor impacto. Quando o Amazon MSK detecta uma falha de agente, ele mitiga a falha ou substitui o agente não íntegro ou inacessível por um novo. Além disso, sempre que possível, ele reutiliza o armazenamento do agente mais antigo para reduzir os dados que o Apache Kafka precisa replicar. Seu impacto na disponibilidade é limitado ao tempo necessário para o Amazon MSK concluir a detecção e a recuperação. Após uma recuperação, os aplicativos de produtor e consumidor podem continuar se comunicando com os mesmos endereços IP do agente usados antes da falha.

Configuração do Amazon MSK

Antes de usar o Amazon MSK pela primeira vez, conclua as seguintes tarefas.

Tarefas

- Inscreva-se para AWS
- Fazer download de bibliotecas e ferramentas

Inscreva-se para AWS

Quando você se inscreve AWS, sua conta da Amazon Web Services é automaticamente cadastrada em todos os serviços AWS, incluindo o Amazon MSK. A cobrança incorrerá apenas pelos serviços utilizados.

Se você já tiver uma AWS conta, vá para a próxima tarefa. Se ainda não possuir uma conta da AWS , use o procedimento a seguir para criar uma.

Para cadastrar uma conta da Amazon Web Services

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

Fazer download de bibliotecas e ferramentas

As seguintes bibliotecas e ferramentas podem ajudar você a trabalhar com o Amazon MSK::

- A AWS Command Line Interface (AWS CLI) é compatível com o Amazon MSK. AWS CLI Isso permite que você controle vários Amazon Web Services a partir da linha de comando e os automatize por meio de scripts. Atualize sua versão AWS CLI para a versão mais recente para

garantir que ela tenha suporte aos recursos do Amazon MSK que estão documentados neste guia do usuário. Para obter instruções detalhadas sobre como atualizar a AWS CLI, consulte Como instalar a AWS Command Line Interface. Depois de instalar o AWS CLI, você deve configurá-lo. Para obter informações sobre como configurar o AWS CLI, consulte aws configure.

- A Referência de API do Amazon Managed Streaming for Kafka documenta as operações de API compatíveis com o Amazon MSK.
- Os Amazon Web Services SDKs para Go, Java, .NET JavaScript, Node.js, PHP, Python e Ruby incluem suporte e exemplos do Amazon MSK.

O que é MSK Provisioned?

Os clusters provisionados do Amazon MSK oferecem uma ampla variedade de recursos e capacidades para ajudá-lo a otimizar o desempenho do seu cluster e atender às suas necessidades de streaming. Os tópicos abaixo descrevem a funcionalidade em detalhes.

O MSK Provisioned é uma opção de implantação de cluster do MSK que permite configurar e escalar manualmente seus clusters do Apache Kafka. Isso fornece níveis variados de controle sobre a infraestrutura que alimenta seu ambiente Apache Kafka. Com o MSK Provisioned, você pode escolher os tipos de instância, os volumes de armazenamento (intermediários padrão) e o número de nós de intermediários que compõem seus clusters do Kafka. Você também pode escalar seu cluster adicionando ou removendo agentes à medida que suas necessidades de processamento de dados evoluem. Essa flexibilidade permite que você otimize os clusters para suas necessidades específicas de carga de trabalho, seja maximizando a taxa de transferência, a capacidade de retenção ou outras características de desempenho. Além das opções de configuração da infraestrutura, o MSK Provisioned oferece benefícios operacionais, de monitoramento e de segurança de nível corporativo. Isso inclui recursos como atualizações de versão do Apache Kafka, segurança integrada por meio de criptografia e controle de acesso e integração com outros, Serviços da AWS como a Amazon, para monitoramento. CloudWatch A MSK Provisioned oferece dois tipos principais de corretores: Standard e Express.

Para obter informações sobre a API provisionada do MSK, consulte a Referência da API do Amazon MSK.

Conceitos básicos sobre como usar o Amazon MSK

Este tutorial mostra um exemplo de como criar um cluster do MSK, produzir e consumir dados e monitorar a integridade do seu cluster usando métricas. Este exemplo não representa todas as opções que você pode escolher ao criar um cluster do MSK. Em diferentes partes deste tutorial, escolhemos as opções padrão para facilitar. Isso não significa que estas são as únicas opções disponíveis para configurar um cluster do MSK ou instâncias de cliente.

Tópicos

Etapa 1: criar um cluster provisionado pelo MSK

Nesta etapa de Introdução ao uso do Amazon MSK, você cria um cluster provisionado do Amazon MSK. Você usa a opção Criação rápida no AWS Management Console para criar esse cluster.

Para criar um cluster Amazon MSK usando o AWS Management Console

1. Faça login no AWS Management Console e abra o console do Amazon MSK em https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. Selecione Criar cluster.
3. Em Método de criação, deixe a opção Criação rápida selecionada. A opção Criação rápida permite criar um cluster com as configurações padrão.
4. Em Nome do cluster, insira um nome descritivo para o cluster. Por exemplo, .**`MSKTutorialCluster`**
5. Para propriedades gerais do cluster, faça o seguinte:
   a. Em Tipo de cluster, escolha Provisionado.
   b. Escolha uma versão do Apache Kafka para ser executada nas corretoras. Escolha Exibir compatibilidade de versões para ver uma tabela comparativa.
   c. Para o tipo de corretor, escolha corretores padrão ou expressos.
   d. Escolha um tamanho de corretor.
6. Na tabela em Todas as configurações de cluster, copie e salve os valores das configurações a seguir, pois você precisará deles posteriormente neste tutorial:
   - VPC
   - Sub-redes
   - Grupos de segurança associados à VPC
7. Selecione Criar cluster.
8. Verifique o Status do cluster na página Resumo do cluster. O status muda de Criando para Ativo conforme o Amazon MSK provisiona o cluster. Quando o status estiver Ativo, você poderá se

conectar ao cluster. Para obter mais informações sobre status de cluster, consulte Entenda os estados do cluster provisionado pelo MSK.

Próxima etapa

Etapa 2: criar um perfil do IAM concedendo acesso para criar tópicos no cluster do Amazon MSK

Etapa 2: criar um perfil do IAM concedendo acesso para criar tópicos no cluster do Amazon MSK

Nesta etapa, você executará duas tarefas. A primeira tarefa será a criação de uma política do IAM que conceda acesso para criar tópicos no cluster e enviar dados para esses tópicos. A segunda tarefa será a criação de um perfil do IAM e a associação dessa política a ele. Em uma etapa posterior, você criará uma máquina cliente que vai assumir esse perfil e usá-lo para criar um tópico no cluster e enviar dados para esse tópico.

Para criar uma política do IAM que permita criar tópicos e gravar neles

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Políticas.
3. Selecione Criar política.
4. No Editor de políticas, escolha JSON e, em seguida, substitua o JSON na janela do editor pelo seguinte JSON.

   No exemplo a seguir, substitua o seguinte:

   - *region*com o código de Região da AWS onde você criou seu cluster.
   - Exemplo de ID da conta*123456789012*, com seu Conta da AWS ID.
   - *MSKTutorialCluster*e*MSKTutorialCluster*/*7d7131e1-25c5-4e9a-9ac5-ea85bee4da11-14*, com o nome do seu cluster e seu 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 obter instruções sobre como escrever políticas seguras, consultethe section called “Controle de acesso do IAM”.

5. Escolha Próximo.
6. Na página Revisar e criar, faça o seguinte:

a. Em Nome da política, insira um nome descritivo, como **`msk-tutorial-policy`**.

b. Em Permissões definidas nesta política, revise e and/or edite as permissões definidas em sua política.

c. (Opcional) Para ajudar a identificar, organizar ou pesquisar a política, escolha Adicionar nova tag para adicionar tags como pares de valores-chave. Por exemplo, adicione uma tag à sua política com o par de **`Environment`** valores-chave e. **`Test`**

   Para obter mais informações sobre o uso de tags, consulte Tags para AWS Identity and Access Management recursos no Guia do usuário do IAM.

7. Selecione Criar política.

Para criar um perfil do IAM e associar a política a ele

1. No painel de navegação, escolha Funções e, em seguida, escolha Criar função.
2. Na página Select trusted entity (Selecionar entidade confiável), faça o seguinte:

   a. Em Tipo de Entidade Confiável, escolha AWS service (Serviço da AWS).

   b. Para Serviço ou caso de uso, escolha EC2.

   c. Em Use case (Caso de uso), escolha EC2.

3. Escolha Próximo.
4. Na página Add permissions (Adicionar permissões), faça o seguinte:

   a. Na caixa de pesquisa em Políticas de permissões, insira o nome da política que você criou anteriormente para este tutorial. Em seguida, escolha a caixa à esquerda do nome da política.

   b. (Opcional) Defina um limite de permissões. Esse é um atributo avançado que está disponível para perfis de serviço, mas não para perfis vinculados ao serviço. Para obter informações sobre como definir um limite de permissões, consulte Como criar funções e anexar políticas (console) no Guia do usuário do IAM.

5. Escolha Próximo.
6. Na página Name, review, and create (Nomear, revisar e criar), faça o seguinte:

   a. Em Nome da função, insira um nome descritivo, como **`msk-tutorial-role`**.

> **Important**
>
> Quando nomear um perfil, observe o seguinte:
>
> - Os nomes das funções devem ser exclusivos dentro de você Conta da AWS e não podem ser diferenciados por maiúsculas e minúsculas.
>
> Por exemplo, não crie dois perfis denominados **`PRODROLE`** e **`prodrole`**. Quando usado em uma política ou como parte de um ARN, o nome de perfil diferencia maiúsculas de minúsculas. No entanto, quando exibido para os clientes no console, como durante o processo de login, o nome de perfil diferencia maiúsculas de minúsculas.
> - Não é possível editar o nome do perfil depois de criá-lo porque outras entidades podem referenciar o perfil.

b. (Opcional) Em Descrição, insira uma descrição para o perfil.

c. (Opcional) Para editar os casos de uso e as permissões da função, na Etapa 1: Selecionar entidades confiáveis ou Etapa 2: Adicionar seções de permissões, escolha Editar.

d. (Opcional) Para ajudar a identificar, organizar ou pesquisar a função, escolha Adicionar nova tag para adicionar tags como pares de valores-chave. Por exemplo, adicione uma tag à sua função com o par de valores-chave de e. **`ProductManager John`**

 Para obter mais informações sobre o uso de tags, consulte Tags para AWS Identity and Access Management recursos no Guia do usuário do IAM.

7. Reveja a função e escolha Criar função.

Próxima etapa

Etapa 3: criar uma máquina cliente

Etapa 3: criar uma máquina cliente

Nesta etapa de Conceitos básicos sobre como usar o Amazon MSK, você criará uma máquina cliente. Use essa máquina cliente para criar um tópico que produza e consuma dados. Para simplificar, você criará essa máquina cliente na VPC associada ao cluster do MSK para que o cliente possa se conectar facilmente ao cluster.

Como criar uma máquina cliente

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel do EC2 console da Amazon, escolha Launch instance.
3. Em Nome e tags, em Nome, insira um nome descritivo para sua máquina cliente para que você possa controlá-la facilmente. Por exemplo, .**`MSKTutorialClient`**
4. Em Imagens do aplicativo e do sistema operacional (Amazon Machine Image), para Amazon Machine Image (AMI), escolha Amazon Linux 2 AMI (HVM) - Kernel 5.10, Tipo de volume SSD.
5. Para Tipo de instância, mantenha a seleção padrão de t2.micro.
6. Em Par de chaves (login), escolha um par de chaves existente ou crie um novo. Se você não precisar de um par de chaves para se conectar à sua instância, você pode escolher Continuar sem um par de chaves (não recomendado).

   Para criar um novo par de chaves, faça o seguinte:

   a. Escolha Criar novo par de chaves.
   b. Em Key pair name (Nome do par de chaves), insira **`MSKKeyPair`**.
   c. Para o tipo de par de chaves e o formato de arquivo de chave privada, mantenha as seleções padrão.
   d. Escolha Create key pair (Criar par de chaves).

   Se preferir, use um par de chaves existente.
7. Role a página para baixo e expanda a seção Detalhes avançados e faça o seguinte:

   - Para o perfil da instância do IAM, escolha uma função do IAM que você deseja que a máquina cliente assuma.

     Se você não tiver uma função do IAM, faça o seguinte:

     i. Escolha Criar novo perfil do IAM.
     ii. Execute as etapas mencionadas na Etapa 2: criar uma função do IAM.
8. Escolha Iniciar instância.
9. Escolha View Instances (Exibir instâncias). Na coluna Grupos de segurança, escolha o grupo de segurança que está associado à sua nova instância. Copie o ID do grupo de segurança e salve-o para usar posteriormente.
10. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.

11. No painel de navegação, escolha Security Groups (Grupos de segurança). Encontre o grupo de segurança cujo ID você salvou em the section called “Criar um cluster”.
12. Na guia Regras de entrada, selecione Editar regras de entrada.
13. Escolha Adicionar regra.
14. Na nova regra, escolha All traffic (Todo o tráfego) na coluna Type (Tipo). No segundo campo da coluna Origem, selecione o grupo de segurança da sua máquina cliente. Esse é o grupo cujo nome você salvou após iniciar a instância da máquina cliente.
15. Selecione Salvar rules. Agora, o grupo de segurança do cluster poderá aceitar o tráfego proveniente do grupo de segurança da máquina cliente.

Próxima etapa

Etapa 4: criar um tópico no cluster do Amazon MSK

Etapa 4: criar um tópico no cluster do Amazon MSK

Nesta etapa de Conceitos básicos sobre como usar o Amazon MSK, você instalará bibliotecas e ferramentas do cliente do Apache Kafka na máquina cliente e criará um tópico.

> **Warning**
>
> Os números de versão do Apache Kafka usados neste tutorial são apenas exemplos. Recomendamos usar a mesma versão do cliente que a versão do cluster do MSK. Uma versão mais antiga do cliente pode não ter certos recursos e correções de erros críticos.

Tópicos

- Determinando a versão do cluster MSK
- Criação de um tópico na máquina cliente

Determinando a versão do cluster MSK

1. Abra o console do Amazon MSK em https://console.aws.amazon.com/msk/.
2. Na barra de navegação, escolha a região em que você criou o cluster MSK.
3. Escolha o cluster MSK.

4. Anote a versão do Apache Kafka usada no cluster.
5. Substitua as ocorrências de números de versão do Amazon MSK neste tutorial pela versão obtida na Etapa 3.

Criação de um tópico na máquina cliente

1. Conecte-se à sua máquina cliente.
   a. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
   b. No painel de navegação, escolha Instâncias. Em seguida, marque a caixa de seleção ao lado do nome da máquina cliente que você criouEtapa 3: criar uma máquina cliente.
   c. Escolha Actions (Ações) e Connect (Conectar-se). Siga as instruções no console para se conectar à sua máquina cliente.
2. Instale o Java e configure a variável de ambiente da versão Kafka.
   a. Instale o Java na máquina cliente executando o comando a seguir.

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

   b. Armazene a versão Kafka do seu cluster MSK na variável de ambiente,`KAFKA_VERSION`, conforme mostrado no comando a seguir. Você precisará dessas informações em toda a configuração.

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

   Por exemplo, se você estiver usando a versão 3.6.0, use o comando a seguir.

```
export KAFKA_VERSION=3.6.0
```

3. Baixe e extraia o Apache Kafka.
   a. Execute o comando a seguir para fazer download do Apache Kafka.

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

> **Note**
>
> A lista a seguir apresenta algumas informações alternativas de download do Kafka que você pode usar se encontrar algum problema.
>
> - Se você encontrar problemas de conectividade ou quiser usar um site espelho, tente usar o seletor de espelhos do Apache, conforme mostrado no comando a seguir.
>
>   ```
>   wget https://www.apache.org/dyn/closer.cgi?path=/kafka/$KAFKA_VERSION/kafka_2.13-$KAFKA_VERSION.tgz
>   ```
>
> - Baixe uma versão apropriada diretamente do site do Apache Kafka.

b. Execute o comando a seguir no diretório onde você fez download do arquivo TAR na etapa anterior.

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

c. Armazene o caminho completo para o diretório recém-criado dentro da variável de `KAFKA_ROOT` ambiente.

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

4. Configure a autenticação para seu cluster MSK.

a. Encontre a versão mais recente da biblioteca de cliente IAM do Amazon MSK. Essa biblioteca permite que sua máquina cliente acesse o cluster MSK usando a autenticação do IAM.

b. Usando os comandos a seguir, navegue até o `$KAFKA_ROOT/libs` diretório e baixe o Amazon MSK IAM JAR associado que você encontrou na etapa anterior. Certifique-se de *`{LATEST VERSION}`* substituir pelo número da versão real que você está baixando.

```
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 executar qualquer comando do Kafka que interaja com seu cluster MSK, talvez seja necessário adicionar o arquivo JAR IAM do Amazon MSK ao seu classpath Java. Defina a variável de `CLASSPATH` ambiente, conforme mostrado no exemplo a seguir.
>
> ```
> export CLASSPATH=$KAFKA_ROOT/libs/aws-msk-iam-auth-{LATEST VERSION}-
> all.jar
> ```
>
> Isso define o `CLASSPATH` para toda a sessão, disponibilizando o JAR para todos os comandos subsequentes do Kafka.

c. Vá até o `$KAFKA_ROOT/config` diretório para criar o arquivo de configuração do cliente.

```
cd $KAFKA_ROOT/config
```

d. Copie e cole as seguintes configurações de propriedade em um novo arquivo. Salve o arquivo como **`client.properties`**.

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

5. (Opcional) Ajuste o tamanho da pilha Java para as ferramentas do Kafka.

   Se você encontrar algum problema relacionado à memória ou estiver trabalhando com um grande número de tópicos ou partições, poderá ajustar o tamanho do heap Java. Para fazer isso, defina a variável de `KAFKA_HEAP_OPTS` ambiente antes de executar os comandos do Kafka.

   O exemplo a seguir define o tamanho máximo e inicial da pilha como 512 megabytes. Ajuste esses valores de acordo com seus requisitos específicos e os recursos disponíveis do sistema.

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

6. Obtenha as informações de conexão do seu cluster.

a. Abra o console do Amazon MSK em https://console.aws.amazon.com/msk/.

b. Aguarde até que o status do seu cluster esteja Ativo. Isso pode demorar vários minutos. Depois que o status ficar Ativo, escolha o nome do cluster. Isso levará você a uma página com o resumo do cluster.

c. Escolha Exibir informações do cliente.

d. Copie a string de conexão para o endpoint privado.

   Você receberá três endpoints para cada um dos corretores. Armazene uma dessas cadeias de conexão na variável de ambienteBOOTSTRAP_SERVER, conforme mostrado no comando a seguir. *<bootstrap-server-string>*Substitua pelo valor real da cadeia de conexão.

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

7. Execute o comando a seguir para criar o tópico.

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

Se você receber um `NoSuchFileException` para o `client.properties` arquivo, certifique-se de que esse arquivo exista no diretório de trabalho atual dentro do diretório bin do Kafka.

> **Note**
>
> Se você preferir não definir a variável de `CLASSPATH` ambiente para toda a sessão, você pode alternativamente prefixar cada comando do Kafka com a variável. `CLASSPATH` Essa abordagem aplica o classpath somente a esse 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 se o tópico foi criado com êxito.

a. Se o comando for bem-sucedido, você deverá ver a seguinte mensagem: `Created topic MSKTutorialTopic.`

b. Liste todos os tópicos para confirmar que seu tópico existe.

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

Se o comando não for bem-sucedido ou se você encontrar um erro, consulte Solução de problemas para o cluster do Amazon MSK para obter informações sobre solução de problemas.

9. (Opcional) Exclua as variáveis de ambiente que você usou neste tutorial.

   Se você quiser manter suas variáveis de ambiente para as próximas etapas deste tutorial, pule esta etapa. Caso contrário, você pode cancelar a definição dessas variáveis, conforme mostrado no exemplo a seguir.

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

Próxima etapa

Etapa 5: produzir e consumir dados

Etapa 5: produzir e consumir dados

Nesta etapa de Conceitos básicos sobre como usar o Amazon MSK, você produzirá e consumirá dados.

Como produzir e consumir mensagens

1. Execute o comando a seguir para iniciar um produtor de console.

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

2. Insira a mensagem que desejar e pressione Enter. Repita esta etapa duas ou três vezes. Toda vez que você inserir uma linha e pressionar Enter, essa linha será enviada para o cluster do Apache Kafka como uma mensagem separada.

3. Mantenha a conexão com a máquina cliente aberta e abra uma segunda conexão separada com esse computador em uma nova janela. Como essa é uma nova sessão, defina as variáveis de BOOTSTRAP_SERVER ambiente KAFKA_ROOT e novamente. Para obter informações sobre como definir essas variáveis de ambiente, consulteCriação de um tópico na máquina cliente.
4. Execute o comando a seguir com sua segunda cadeia de conexão com a máquina cliente para criar um consumidor de console.

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

   Você deve começar a ver as mensagens inseridas anteriormente ao usar o comando console producer.
5. Insira mais mensagens na janela do produtor e observe-as aparecerem na janela do consumidor.

Próxima etapa

Etapa 6: Use CloudWatch a Amazon para visualizar as métricas do Amazon MSK

Etapa 6: Use CloudWatch a Amazon para visualizar as métricas do Amazon MSK

Nesta etapa de Introdução ao uso do Amazon MSK, você analisa as métricas do Amazon MSK na Amazon. CloudWatch

Para visualizar as métricas do Amazon MSK em CloudWatch

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. Escolha a guia All metrics (Todas as métricas) e escolha AWS/Kafka.
4. Para visualizar métricas no nível de agente, escolha Broker ID, Cluster Name (ID do agente, Nome do cluster). Para métricas no nível de cluster, escolha Cluster Name (Nome do cluster).
5. (Opcional) No painel gráfico, selecione uma estatística e um período de tempo e, em seguida, crie um CloudWatch alarme usando essas configurações.

Próxima etapa

Etapa 7: Excluir os AWS recursos criados para este tutorial

Etapa 7: Excluir os AWS recursos criados para este tutorial

Na etapa final de Conceitos básicos sobre como usar o Amazon MSK, você exclui o cluster do MSK e a máquina cliente que criou para este tutorial.

Para excluir os recursos usando o AWS Management Console

1. Abra o console do Amazon MSK em https://console.aws.amazon.com/msk/.
2. Selecione o nome do seu cluster. Por exemplo, MSKTutorialCluster.
3. Escolha Actions (Ações) e Delete (Excluir).
4. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
5. Escolha a instância que você criou para sua máquina cliente, por exemplo, **`MSKTutorialClient`**.
6. Escolha Estado da instância e Encerrar instância.

Para excluir a política e o perfil do IAM

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Perfis.
3. Na caixa de pesquisa, insira o nome do perfil do IAM que você criou para este tutorial.
4. Selecione o perfil de . Escolha Excluir perfil e confirme a exclusão.
5. No painel de navegação, escolha Políticas.
6. Na caixa de pesquisa, insira o nome da política que você criou para este tutorial.
7. Escolha a política para abrir a respectiva página de resumo. Na página Resumo da política, escolha Editar política.
8. Escolha Excluir.

Amazon MSK: funcionamento

O Amazon MSK é um serviço Apache Kafka totalmente gerenciado que facilita a criação e a execução de aplicativos que usam o Apache Kafka para processar dados de streaming. Este guia

fornece informações para ajudar os desenvolvedores a entender como o Amazon MSK funciona e como usá-lo de forma eficaz em seus aplicativos.

Em um alto nível, o Amazon MSK fornece um cluster Apache Kafka totalmente gerenciado que é provisionado e operado pela. AWS Isso significa que você não precisa se preocupar em provisionar EC2 instâncias, definir configurações de rede, gerenciar corretores Kafka ou realizar tarefas de manutenção contínuas. Em vez disso, você pode se concentrar na criação de seu aplicativo e deixar que a Amazon MSK cuide da infraestrutura. O Amazon MSK provisiona automaticamente os recursos de computação, armazenamento e rede necessários e fornece recursos como escalabilidade automática, alta disponibilidade e failover para garantir que seu cluster Kafka seja confiável e altamente disponível. Este guia aborda os principais componentes do Amazon MSK e como você pode usá-lo para criar aplicativos de streaming de dados.

Gerencie seu cluster provisionado

Um cluster do Amazon MSK é o recurso primário do Amazon MSK que você pode criar em sua conta. Os tópicos desta seção descrevem como realizar operações comuns do Amazon MSK. Para obter uma lista de todas as operações que você pode realizar em um cluster do MSK, consulte:

- A AWS Management Console
- A Referência de API do Amazon MSK
- A Referência de comandos da CLI do Amazon MSK

Tópicos

- Atualizar o tamanho do agente de cluster do Amazon MSK
- Use o LinkedIn Cruise Control para Apache Kafka com o Amazon MSK
- Atualizar a configuração de um cluster do Amazon MSK
- Reinicializar um agente de um cluster do Amazon MSK
- Marcar um cluster do Amazon MSK
- Migrar para um cluster do Amazon MSK
- Excluir um cluster provisionado do Amazon MSK

Crie um cluster provisionado pelo MSK

> **Important**
>
> Você não pode alterar a VPC de um cluster provisionado pelo MSK depois de criar o cluster.

Antes de criar um cluster provisionado pelo MSK, você precisa ter uma (Amazon Virtual Private Cloud VPC) e configurar sub-redes dentro dessa VPC.

Para corretores padrão na região Oeste dos EUA (Norte da Califórnia), você precisa de duas sub-redes em duas zonas de disponibilidade diferentes. Em todas as outras regiões que disponibilizam o Amazon MSK, é possível especificar duas ou três sub-redes. As suas sub-redes devem estar em diferentes zonas de disponibilidade. Para corretores Express, você precisa de três sub-redes em três zonas de disponibilidade diferentes. Quando você cria um cluster provisionado pelo MSK, o Amazon MSK distribui os nós do agente uniformemente pelas sub-redes que você especifica.

Tópicos

- Crie um cluster provisionado pelo MSK usando o AWS Management Console
- Crie um cluster Amazon MSK provisionado usando o AWS CLI
- Crie um cluster provisionado pelo MSK com uma configuração personalizada do Amazon MSK usando o AWS CLI
- Crie um cluster provisionado pelo MSK usando a API do Amazon MSK

Crie um cluster provisionado pelo MSK usando o AWS Management Console

Os procedimentos neste tópico descrevem a tarefa comum de criar um cluster provisionado pelo MSK usando a opção de criação personalizada no. AWS Management Console Usando outras opções disponíveis no AWS Management Console, você também pode criar o seguinte:

- Um cluster sem servidor
- Um cluster provisionado pelo MSK usando a opção de criação rápida

Procedimentos neste tópico

- Etapa 1: instalação e configuração iniciais do cluster
- Etapa 2: Definir as configurações de armazenamento e cluster
- Etapa 3: definir as configurações de rede
- Etapa 4: definir as configurações de segurança
- Etapa 5: configurar as opções de monitoramento
- Etapa 6: revisar a configuração do cluster

Etapa 1: instalação e configuração iniciais do cluster

1. Abra o console do Amazon MSK em https://console.aws.amazon.com/msk/.
2. Selecione Criar cluster.
3. Em Método de criação de cluster, escolha Criação personalizada.
4. Em Nome do cluster, especifique um nome que seja exclusivo e contenha no máximo 64 caracteres.
5. Em Tipo de cluster, escolha Provisionado.
6. Para a versão Apache Kafka, escolha uma versão para ser executada nos corretores. Para ver uma comparação dos recursos do Amazon MSK que são compatíveis com cada versão do Apache Kafka, escolha Exibir compatibilidade de versão.
7. Na seção Corretores, faça o seguinte:

    a. Para o tipo de corretor, escolha uma das seguintes opções:

- Corretores expressos: corretores escaláveis e de alto desempenho com armazenamento virtual totalmente gerenciado. Escolha esse tipo de corretor para aplicativos exigentes e de alto rendimento.
- Corretores padrão: corretor Kafka tradicional com controle total de configuração. Escolha esse tipo de agente para cargas de trabalho de uso geral com requisitos moderados de taxa de transferência.

Para obter mais informações sobre esses tipos de corretores, consulteTipos de corretores Amazon MSK.

b. Para o tamanho do Broker, escolha um tamanho a ser usado para o cluster com base nas necessidades de computação, memória e armazenamento do cluster.

c. Em Número de zonas, escolha o número de corretores Zonas de disponibilidade da AWSnos quais os corretores são distribuídos.

Os corretores expressos exigem três zonas de disponibilidade para maior disponibilidade.

d. Para corretores por zona, especifique o número de corretores que você deseja que o Amazon MSK crie em cada zona de disponibilidade. O mínimo é um agente por zona de disponibilidade e o máximo é 30 agentes por cluster para clusters ZooKeeper baseados e 60 corretores por cluster para clusters KRaftbaseados.

Etapa 2: Definir as configurações de armazenamento e cluster

Este procedimento descreve como você pode configurar suas necessidades de armazenamento de dados em todos os corretores e especificar o modo de armazenamento. Isso ajuda você a definir seus requisitos de armazenamento de dados com base em suas necessidades de carga de trabalho. Além disso, esse procedimento descreve as configurações do cluster que controlam a forma como seus agentes operam. Essas configurações incluem configurações de intermediário, configurações de tópicos padrão e política de armazenamento hierárquico.

1. Se você selecionou o tipo de corretor como Padrão, faça o seguinte na seção Armazenamento:

   a. Em Armazenamento, escolha a quantidade inicial de armazenamento que você deseja que seu cluster tenha. Não é possível diminuir a capacidade de armazenamento depois de criar o cluster.

   b. (Opcional) Dependendo do tamanho do agente (tamanho da instância) selecionado, você também pode especificar a taxa de transferência de armazenamento provisionado por

agente. Essa opção permite que você aloque desempenho dedicado de entrada e saída (E/S) para os volumes do Amazon EBS de cada agente.

Para habilitar essa opção, escolha o tamanho do agente (tamanho da instância) kafka.m5.4xlarge ou maior para x86 e kafka.m7g.2xlarge ou maior para instâncias baseadas no Graviton. Em seguida, escolha a caixa de seleção Habilitar taxa de transferência de armazenamento provisionado. Ao marcar essa caixa de seleção, você pode definir manualmente um mínimo de 250 MiB por segundo de taxa de transferência. Isso é útil para cargas de trabalho com uso intenso de E/S ou aplicativos que exigem desempenho de armazenamento previsível e de alta velocidade. Para obter mais informações, consulte ???.

c. Para o modo de armazenamento em cluster, especifique como os dados são armazenados e gerenciados em seu cluster. Essa opção determina o tipo e a configuração do armazenamento usado por seus corretores. Escolha uma das seguintes opções:

- Somente armazenamento do EBS: armazena todos os dados de tópicos localmente nos volumes do Amazon Elastic Block Store (Amazon EBS) anexados a cada corretor. Escolha esse modo para necessidades de desempenho consistentes e acesso rápido às mensagens recentes.
- Armazenamento hierárquico e armazenamento EBS: combina dados locais do Amazon EBS com armazenamento remoto e econômico para grandes conjuntos de dados no Amazon S3. Esse modo reduz os custos de armazenamento do Amazon EBS, suporta maior retenção de dados e escala o armazenamento automaticamente sem intervenção manual. Escolha esse modo quando quiser reter dados por períodos mais longos a um custo menor ou esperar que suas necessidades de armazenamento aumentem significativamente.

> **Note**
>
> Você não precisa gerenciar o armazenamento para corretores Express.

2. Para a configuração do cluster, especifique uma das seguintes opções para definir o comportamento do seu cluster:

- Configuração padrão do Amazon MSK: contém um conjunto predefinido de configurações otimizadas para casos de uso geral. Escolha essa opção para configuração e implantação rápidas do cluster. Para obter informações sobre configurações do Amazon MSK, consulte Configuração provisionada do Amazon MSK.

- Configuração personalizada: permite que você especifique suas próprias configurações de corretor e tópico. Você pode escolher uma configuração personalizada existente na lista ou criar uma nova configuração personalizada. Escolha essa opção para um controle aprimorado para seus corretores, como ajuste específico de desempenho, configurações de segurança e muito mais.

3. Escolha Próximo para continuar.

Etapa 3: definir as configurações de rede

A configuração de rede define como seu cluster é implantado em sua AWS infraestrutura. Isso inclui VPC, zonas de disponibilidade e sub-redes e grupos de segurança que controlam a rede, a disponibilidade e o acesso.

1. Para redes, faça o seguinte:

   a. Escolha a VPC que você deseja usar para o cluster.

   b. Com base no número de zonas de disponibilidade que você selecionou anteriormente, especifique as zonas de disponibilidade e as sub-redes nas quais os corretores serão implantados.

      Para corretores padrão na região Oeste dos EUA (Norte da Califórnia), você precisa de duas sub-redes em duas zonas de disponibilidade diferentes. Em todas as outras regiões que disponibilizam o Amazon MSK, é possível especificar duas ou três sub-redes. As suas sub-redes devem estar em diferentes zonas de disponibilidade.

      Para corretores Express, você precisa de três sub-redes em três zonas de disponibilidade diferentes.

      Quando você cria um cluster provisionado pelo MSK, o MSK distribui os nós do broker uniformemente pelas sub-redes que você especifica.

   c. Para grupos de segurança na Amazon EC2, escolha ou crie um ou mais grupos de segurança aos quais você deseja dar acesso ao seu cluster. Esses grupos EC2 de segurança da Amazon controlam o tráfego de entrada e saída para seus corretores. Por exemplo, os grupos de segurança das máquinas clientes.

Se especificar grupos de segurança que foram compartilhados com você, deverá garantir que tenha as permissões para usá-los. Especificamente, você precisa da permissão `ec2:DescribeSecurityGroups`. Para obter mais informações, consulte Conectando-se a um cluster MSK.

2. Escolha Próximo para continuar.

Etapa 4: definir as configurações de segurança

1. Na seção Configurações de segurança, faça o seguinte:

   - Escolha um ou mais dos seguintes métodos de autenticação e autorização para controlar o acesso do cliente aos seus clusters do Kafka:

     - Acesso não autenticado: permite que os clientes acessem o cluster sem fornecer nenhuma credencial de autenticação. Esse método é um risco de segurança e pode não estar em conformidade com as melhores práticas de segurança. Para obter mais informações, consulte msk-unrestricted-access-check.
     - Autenticação baseada em funções do IAM: permite a autenticação e autorização do cliente usando usuários/funções AWS do IAM. Esse método fornece controle refinado sobre o acesso ao cluster por meio de políticas do IAM. Recomendamos esse método para aplicativos que já estão sendo executados em AWS.
     - Autenticação SASL/SCRAM: exige que os clientes forneçam credenciais de nome de usuário e senha armazenadas para autenticação. AWS Secrets Manager O Amazon MSK recupera essas credenciais do Secrets Manager e autentica os usuários com segurança.

       Para configurar as credenciais de login relacionadas à autenticação de um cluster, primeiro crie um recurso secreto no Secrets Manager. Em seguida, associe as credenciais de login a esse segredo. Para obter mais informações sobre esse método de controle de acesso, consulteConfigurar a SASL/SCRAM autenticação para um cluster Amazon MSK.
     - Autenticação de cliente TLS por meio de AWS Certificate Manager (ACM): permite a autenticação mútua entre clientes e corretores usando certificados digitais. Você deve configurar um AWS Private Certificate Authority (AWS Private CA) igual ou diferente Conta da AWS do seu cluster.

É altamente recomendável usar AWS Private CA s independentes para cada cluster MSK ao implementar mTLS. Isso garante que os certificados TLS assinados por sejam autenticados PCAs apenas com um único cluster MSK, mantendo assim um controle de acesso rigoroso.

2. Em Criptografia, escolha o tipo de chave KMS que você deseja usar para criptografar dados em repouso. Para obter mais informações, consulte the section called “Criptografia do Amazon MSK em repouso”.

   A criptografia de dados em repouso protege a integridade dos dados armazenados, enquanto a criptografia em trânsito protege a confidencialidade dos dados do monitoramento da rede durante a transferência.

3. Escolha Próximo para continuar.

Etapa 5: configurar as opções de monitoramento

Este procedimento descreve como configurar suas métricas de corretor e coletar e entregar registros do corretor. Com essas configurações, você pode observar e analisar a integridade, o desempenho e solucionar problemas do seu cluster. Para obter mais informações, consulte the section called “Monitorar um cluster”.

1. Para CloudWatch as métricas da Amazon para esse cluster, escolha um dos seguintes níveis de monitoramento. As métricas coletadas em cada nível de monitoramento são integradas CloudWatch para visualização e alertas.

   a. Monitoramento básico: fornece um conjunto de métricas essenciais em nível de cluster sem custo adicional. Esse nível é bom para a maioria dos casos de uso com necessidades gerais de monitoramento.

   b. Monitoramento aprimorado em nível de corretor: fornece métricas detalhadas do corretor a um custo adicional. Esse nível inclui monitoramento básico e métricas de corretoras mais granulares, como métricas de armazenamento hierárquico, bytes in/out de outras corretoras e tempo total de operações. read/write Você paga pelas métricas nesse nível, enquanto as métricas do nível básico continuam sendo gratuitas.

   c. Monitoramento aprimorado em nível de tópico: fornece métricas para tópicos individuais a um custo adicional. Escolha esse nível para obter uma visão mais granular do desempenho do tópico em todos os corretores. Esse nível inclui monitoramento aprimorado em nível de

corretor e métricas em nível de tópico, como métricas de armazenamento hierárquico para um tópico específico e número de mensagens recebidas por segundo.

d. Monitoramento aprimorado em nível de partição: fornece a visão mais granular das métricas por partição a um custo adicional. Escolha esse nível para obter o monitoramento mais detalhado capturando métricas para cada partição em cada tópico entre os corretores. Esse nível inclui monitoramento aprimorado em nível de tópico e métricas específicas de partição refinadas, como métricas de defasagem de compensação.

Para obter mais informações sobre as métricas disponíveis para os tipos de corretores Standard e Express em cada um desses níveis de monitoramento, consulte CloudWatch métricas para corretores padrão CloudWatch métricas para corretores Express e.

2. (Opcional) Se você quiser exportar métricas no formato Prometheus usando JMX Exporter, Node Exporter ou ambos, escolha Habilitar monitoramento aberto com o Prometheus. Para obter mais informações sobre essa opção, consulte Monitorare com o Prometheus.
3. (Opcional) Para configurar seu cluster MSK para fornecer registros de agentes a vários Serviços da AWS para solução de problemas e auditoria, escolha uma ou mais das opções a seguir. O Amazon MSK não cria esses recursos de destino para você se eles ainda não existirem. Para obter mais informações, consulte Logs do agente.

   - Entregue para a Amazon CloudWatch Logs: envia registros para CloudWatch com recursos de agrupamento, pesquisa e visualização. Você pode consultar e analisar registros sem sair do AWS Management Console.
   - Entrega para o Amazon S3: armazena registros como arquivos em buckets do Amazon S3 para arquivamento a longo prazo e análise em lote.
   - Entregue para o Amazon Data Firehose: envie registros para o Firehose para entrega automática ao Amazon OpenSearch Service para solução de problemas em tempo real.
4. (Opcional) Para ajudar a identificar, organizar ou pesquisar seu cluster, escolha Adicionar nova tag para adicionar tags como pares de valores-chave. Por exemplo, adicione uma tag ao seu cluster com o par de **`Load testing`** valores-chave e. **`Test`**

   Para obter mais informações sobre o uso de tags em seus clusters, consulteMarcar um cluster do Amazon MSK.
5. Escolha Próximo para continuar.

Etapa 6: revisar a configuração do cluster

1. Revise as configurações do cluster.

   Escolha Editar ou Anterior para alterar qualquer uma das configurações especificadas anteriormente ou voltar para a tela anterior do console.

2. Selecione Criar cluster.

3. Verifique o status desse cluster na seção Resumo do cluster da página de detalhes do cluster. O status muda de Criando para Ativo conforme o Amazon MSK provisiona o cluster. Quando o status estiver Ativo, você poderá se conectar ao cluster. Para obter mais informações sobre status de cluster, consulte Entenda os estados do cluster provisionado pelo MSK.

Crie um cluster Amazon MSK provisionado usando o AWS CLI

1. Copie o seguinte JSON e salve-o em um arquivo. Nomeie o arquivo `brokernodegroupinfo.json`. Substitua a sub-rede IDs no JSON pelos valores que correspondem às suas sub-redes. As sub-redes devem estar em zonas de disponibilidade diferentes. *"Security-Group-ID"*Substitua pelo ID de um ou mais grupos de segurança da VPC do cliente. Os clientes associados a esses grupos de segurança têm acesso ao cluster. Se você especificar grupos de segurança que foram compartilhados com você, deverá garantir que você tenha permissões para eles. Especificamente, você precisa da permissão `ec2:DescribeSecurityGroups`. Por exemplo, consulte Amazon EC2: Permite gerenciar grupos de EC2 segurança da Amazon associados a uma VPC específica, programaticamente e no console. Por fim, salve o arquivo JSON atualizado no computador em que você o AWS CLI instalou.

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

> **Important**
>
> Para corretores Express, você precisa de três sub-redes em três zonas de disponibilidade diferentes. Você também não precisa definir nenhuma propriedade relacionada ao armazenamento.
> Para corretores padrão na região Oeste dos EUA (Norte da Califórnia), você precisa de duas sub-redes em duas zonas de disponibilidade diferentes. Em todas as outras regiões que disponibilizam o Amazon MSK, é possível especificar duas ou três sub-redes. As suas sub-redes devem estar em diferentes zonas de disponibilidade. Quando você cria um cluster, o Amazon MSK distribui os nós de agente uniformemente pelas sub-redes especificadas.

2. Execute o AWS CLI comando a seguir no diretório em que você salvou o `brokernodegroupinfo.json` arquivo, *`"Your-Cluster-Name"`* substituindo-o por um nome de sua escolha. Para *`"Monitoring-Level"`*, você pode especificar um dos três valores a seguir: `DEFAULT`, `PER_BROKER`, ou `PER_TOPIC_PER_BROKER`. Para obter informações sobre esses três níveis diferentes de monitoramento, consulte ???. O parâmetro `enhanced-monitoring` é opcional. Se não especificá-lo no comando `create-cluster`, você obterá o nível de monitoramento `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"
```

A saída do comando é semelhante ao JSON a seguir:

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

> **Note**
>
> O comando `create-cluster` pode retornar um erro informando que uma ou mais sub-redes pertencem a zonas de disponibilidade que não têm suporte. Quando isso acontece, o erro indica as zonas de disponibilidade que não têm suporte. Crie sub-redes

que não usem as zonas de disponibilidade sem suporte e tente o comando `create-cluster` novamente.

3. Salve o valor da chave `ClusterArn` porque você precisará dele para executar outras ações no cluster.
4. Execute o seguinte comando para verificar o `STATE` do seu cluster. O valor de `STATE` muda de `CREATING` para `ACTIVE` conforme o Amazon MSK provisiona o cluster. Quando o estado for `ACTIVE`, você poderá se conectar ao cluster. Para obter mais informações sobre status de cluster, consulte Entenda os estados do cluster provisionado pelo MSK.

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

Crie um cluster provisionado pelo MSK com uma configuração personalizada do Amazon MSK usando o AWS CLI

Para obter informações sobre configurações personalizadas do Amazon MSK e como criá-las, consulte the section called “Configuração do corretor”.

1. Salve o seguinte JSON em um arquivo, *`configuration-arn`* substituindo-o pelo ARN da configuração que você deseja usar para criar o cluster.

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

2. Execute o comando `create-cluster` e use a opção `configuration-info` para apontar para o arquivo JSON que você salvou na etapa anterior. Veja um exemplo a seguir.

```
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
```

Veja a seguir um exemplo de uma resposta bem-sucedida após a execução desse comando.

```
{
```

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

Crie um cluster provisionado pelo MSK usando a API do Amazon MSK

A API do Amazon MSK permite que você crie e gerencie programaticamente seu cluster MSK Provisioned como parte de scripts automatizados de provisionamento ou implantação de infraestrutura.

Para criar um cluster provisionado pelo MSK usando a API, consulte. CreateCluster

Listar clusters do Amazon MSK

Para obter um agente de bootstrap para um cluster do Amazon MSK, você precisa do nome do recurso da Amazon (ARN) do cluster. Se você não tiver o ARN do cluster, poderá encontrá-lo listando todos os clusters. Consulte the section called “Obtenha os corretores de bootstrap”.

Tópicos

- Listar clusters usando o AWS Management Console
- Listar clusters usando o AWS CLI
- Listar clusters usando a API

Listar clusters usando o AWS Management Console

Para obter um agente de bootstrap para um cluster do Amazon MSK, você precisa do nome do recurso da Amazon (ARN) do cluster. Se você não tiver o ARN do cluster, poderá encontrá-lo listando todos os clusters. Consulte the section called “Obtenha os corretores de bootstrap”.

1. Faça login no AWS Management Console e abra o console do Amazon MSK em https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. A tabela mostra todos os clusters da região atual nesta conta. Escolha o nome de um cluster para visualizar seus detalhes.

Listar clusters usando o AWS CLI

Para obter um agente de bootstrap para um cluster do Amazon MSK, você precisa do nome do recurso da Amazon (ARN) do cluster. Se você não tiver o ARN do cluster, poderá encontrá-lo listando todos os clusters. Consulte the section called “Obtenha os corretores de bootstrap”.

```
aws kafka list-clusters
```

Listar clusters usando a API

Para obter um agente de bootstrap para um cluster do Amazon MSK, você precisa do nome do recurso da Amazon (ARN) do cluster. Se você não tiver o ARN do cluster, poderá encontrá-lo listando todos os clusters. Consulte the section called “Obtenha os corretores de bootstrap”.

Para listar clusters usando a API, consulte ListClusters.

Conecte-se a um cluster provisionado do Amazon MSK

Por padrão, os clientes podem acessar um cluster provisionado pelo MSK somente se estiverem na mesma VPC do cluster. Toda comunicação entre seus clientes Kafka e seu cluster MSK Provisioned é privada por padrão e seus dados de streaming nunca atravessam a Internet. Para se conectar ao seu cluster provisionado pelo MSK a partir de um cliente que está na mesma VPC do cluster, certifique-se de que o grupo de segurança do cluster tenha uma regra de entrada que aceite o tráfego do grupo de segurança do cliente. Para obter informações sobre como configurar essas regras, consulte Regras do grupo de segurança. Para ver um exemplo de como acessar um cluster a partir de uma EC2 instância da Amazon que está na mesma VPC do cluster, consulte. the section called “Conceitos básicos”

> **Note**
>
> KRaft o modo de metadados e os corretores MSK Express não podem ter o monitoramento aberto e o acesso público habilitados.

Para se conectar ao seu cluster provisionado pelo MSK a partir de um cliente que está fora da VPC do cluster, consulte Acesso de dentro, mas de AWS fora da VPC do cluster.

Tópicos

- Ativar o acesso público a um cluster provisionado pelo MSK
- Acesso de dentro AWS , mas de fora da VPC do cluster

Ativar o acesso público a um cluster provisionado pelo MSK

O Amazon MSK oferece a opção de ativar o acesso público aos corretores de clusters provisionados pelo MSK que executam o Apache Kafka 2.6.0 ou versões posteriores. Por motivos de segurança, você não pode ativar o acesso público ao criar um cluster do MSK. No entanto, você pode atualizar um cluster existente para torná-lo acessível ao público. Você pode criar um novo cluster e atualizá-lo para torná-lo acessível publicamente.

Você pode ativar o acesso público a um cluster MSK sem custo adicional, mas os custos padrão de transferência de AWS dados se aplicam à transferência de dados para dentro e para fora do cluster. Para obter informações sobre preços, consulte Amazon EC2 On-Demand Pricing.

> **Note**
>
> Se você estiver usando os métodos de controle de acesso SASL/SCRAM ou mTLS, você deve primeiro configurar o Apache Kafka para seu cluster. ACLs Em seguida, atualize a configuração do cluster para definir a `allow.everyone.if.no.acl.found` propriedade como false. Para obter informações sobre como atualizar a configuração de um cluster, consulte the section called “Operações de configuração do broker”.

Para ativar o acesso público a um cluster provisionado pelo MSK, certifique-se de que o cluster atenda a todas as seguintes condições:

- As sub-redes associadas ao cluster devem ser públicas. Cada sub-rede pública tem um IPv4 endereço público associado a ela, e os preços dos IPv4 endereços públicos são mostrados na página de preços da Amazon VPC. Isso significa que as sub-redes devem ter uma tabela de rotas associada a um gateway da Internet conectado. Para obter informações sobre como criar e conectar um gateway de internet, consulte Habilitar o acesso VPC à internet usando gateways de internet no Guia do usuário da Amazon VPC.
- O controle de acesso não autenticado deve estar desativado e pelo menos um dos seguintes métodos de controle de acesso deve estar ativado:, mTLS. SASL/IAM, SASL/SCRAM Para obter informações sobre como atualizar o método de controle de acesso de um cluster, consulte the section called “Atualize a segurança do cluster”.

- A criptografia dentro do cluster deve estar ativada. A configuração ativada é o padrão ao criar um cluster. Não é possível ativar a criptografia dentro do cluster para um cluster que tenha sido criado com ela desativada. Portanto, não é possível ativar o acesso público para um cluster que tenha sido criado com a criptografia no cluster desativada.
- O tráfego de texto simples entre agentes e clientes deve estar desativado. Para obter informações sobre como desativá-lo se estiver ativado, consulte the section called “Atualize a segurança do cluster”.
- Se você estiver usando o controle de acesso do IAM e quiser aplicar políticas de autorização ou atualizar suas políticas de autorização, consultethe section called “Controle de acesso do IAM”. Para obter informações sobre o Apache Kafka ACLs, consulte. the section called “Apache Kafka ACLs”

Depois de garantir que um cluster MSK atenda às condições listadas acima, você pode usar a API AWS Management Console AWS CLI, a ou a Amazon MSK para ativar o acesso público. Depois de ativar o acesso público a um cluster, você pode obter uma string pública de agentes de bootstrap para ele. Para obter informações sobre a obtenção de agentes de bootstrap para um cluster, consulte the section called “Obtenha os corretores de bootstrap”.

Important

Além de ativar o acesso público, certifique-se de que os grupos de segurança do cluster tenham regras de TCP de entrada que permitam acesso público do seu endereço IP. Recomendamos tornar essas regras o mais restritivas possível. Para obter mais informações sobre grupos de segurança e regras de entrada, consulte Grupos de segurança para sua VPC no Guia do usuário da Amazon VPC. Para obter os números das portas, consulte the section called “Informações de porta”. Para obter instruções sobre como alterar o grupo de segurança de um cluster, consulte the section called “Alterar os grupos de segurança”.

Note

Se você usar as instruções a seguir para ativar o acesso público e ainda não conseguir acessar o cluster, consulte the section called “Não é possível acessar o cluster que está com o acesso público ativado”.

Ativar o acesso público usando o console

1. Faça login no AWS Management Console e abra o console do Amazon MSK em https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. Na lista de clusters, selecione o cluster ao qual deseja ativar o acesso público.
3. Escolha a guia Propriedades e, em seguida, encontre a seção Configurações de rede.
4. Escolha Editar acesso público.

Ativando o acesso público usando o AWS CLI

1. Execute o AWS CLI comando a seguir, substituindo *ClusterArn* e *Current-Cluster-Version* pelo ARN e pela versão atual do cluster. Para encontrar a versão atual do cluster, use a DescribeClusteroperação ou o comando AWS CLI describe-cluster. Uma versão de exemplo é KTVPDKIKX0DER.

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

A saída desse comando update-connectivity é semelhante ao seguinte JSON de exemplo.

```
{
    "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 desativar o acesso público, use um AWS CLI comando semelhante, mas com as seguintes informações de conectividade:
>
> ```
> '{"PublicAccess": {"Type": "DISABLED"}}'
> ```

2. Para obter o resultado da `update-connectivity` operação, execute o comando a seguir, *`ClusterOperationArn`* substituindo-o pelo ARN obtido na saída do `update-connectivity` comando.

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

A saída desse comando `describe-cluster-operation` é semelhante ao seguinte JSON de exemplo.

```
{
    "ClusterOperationInfo": {
        "ClientRequestId": "982168a3-939f-11e9-8a62-538df00285db",
        "ClusterArn": "arn:aws:kafka:us-east-1:012345678012:cluster/
exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2",
        "CreationTime": "2019-06-20T21:08:57.735Z",
        "OperationArn": "arn:aws:kafka:us-east-1:012345678012:cluster-
operation/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2/0123abcd-
abcd-4f7f-1234-9876543210ef",
        "OperationState": "UPDATE_COMPLETE",
        "OperationType": "UPDATE_CONNECTIVITY",
        "SourceClusterInfo": {
            "ConnectivityInfo": {
                "PublicAccess": {
                    "Type": "DISABLED"
                }
            }
        },
        "TargetClusterInfo": {
            "ConnectivityInfo": {
                "PublicAccess": {
                    "Type": "SERVICE_PROVIDED_EIPS"
                }
            }
        }
    }
}
```

Se `OperationState` tiver o valor `UPDATE_IN_PROGRESS`, aguarde um pouco e execute o comando `describe-cluster-operation` novamente.

Ativar o acesso público usando a API do Amazon MSK

- Para usar a API para ativar ou desativar o acesso público a um cluster, consulte UpdateConnectivity.

> **Note**
>
> Por motivos de segurança, o Amazon MSK não permite acesso público aos nós do Apache ZooKeeper ou do KRaft controlador.

Acesso de dentro AWS , mas de fora da VPC do cluster

Para se conectar a um cluster MSK de dentro AWS , mas de fora da Amazon VPC do cluster, existem as seguintes opções.

Emparelhamento do Amazon VPC

Para se conectar ao seu cluster MSK a partir de uma VPC diferente da VPC do cluster, você pode criar uma conexão de emparelhamento entre as duas. VPCs Para obter informações sobre o emparelhamento da VPC, consulte Guia de emparelhamento do Amazon VPC.

AWS Direct Connect

AWS Direct Connect conecta sua rede local a AWS mais de um cabo de fibra óptica Ethernet padrão de 1 gigabit ou 10 gigabit. Uma extremidade do cabo está conectada ao roteador e a outra ao AWS Direct Connect roteador. Com essa conexão estabelecida, você pode criar interfaces virtuais diretamente na AWS nuvem e na Amazon VPC, ignorando os provedores de serviços de Internet em seu caminho de rede. Para obter mais informações, consulte AWS Direct Connect.

AWS Transit Gateway

AWS Transit Gateway é um serviço que permite conectar sua rede VPCs e sua rede local a um único gateway. Para obter informações sobre como usar o AWS Transit Gateway, consulte AWS Transit Gateway.

Conexões da VPN

É possível conectar a VPC do cluster do MSK a redes remotas e usuários que usam as opções de conectividade por VPN descritas no seguinte tópico: Conexões por VPN.

Proxies REST

É possível instalar um proxy REST em uma instância sendo executada na Amazon VPC do cluster. Os proxies REST permitem que os produtores e os consumidores se comuniquem com o cluster por meio de solicitações da API do HTTP.

Conectividade multi-VPC em múltiplas regiões

O documento a seguir descreve as opções de conectividade para várias VPCs que residem em regiões diferentes: Conectividade multi-VPC de várias regiões.

Conectividade privada multi-VPC de região única

A conectividade privada de várias VPCs (desenvolvida por AWS PrivateLink) para clusters Amazon Managed Streaming for Apache Kafka (Amazon MSK) é um recurso que permite conectar mais rapidamente clientes Kafka hospedados em diferentes VPCs nuvens privadas virtuais () e contas a um cluster Amazon MSK. AWS

Consulte Conectividade multi-VPC de região única para clientes entre contas.

EC2- A rede clássica foi descontinuada

O Amazon MSK não oferece mais suporte a EC2 instâncias da Amazon em execução com a rede Amazon EC2 -Classic.

Veja EC2- A rede clássica está se aposentando — Veja como se preparar.

Conectividade privada multi-VPC do Amazon MSK em uma única região

A conectividade privada de várias VPCs (desenvolvida por AWS PrivateLink) para clusters Amazon Managed Streaming for Apache Kafka (Amazon MSK) é um recurso que permite conectar mais rapidamente clientes Kafka hospedados em diferentes VPCs nuvens privadas virtuais () e contas a um cluster Amazon MSK. AWS

A conectividade privada multi-VPC é uma solução gerenciada que simplifica a infraestrutura de rede para conectividade multi-VPCs e entre contas. Os clientes podem se conectar ao cluster Amazon MSK PrivateLink sem deixar de manter todo o tráfego na AWS rede. A conectividade privada de várias VPCs para clusters do Amazon MSK está disponível em todas as regiões em AWS que o Amazon MSK está disponível.

Tópicos

- O que é conectividade privada multi-VPC?

O que é conectividade privada multi-VPC?

A conectividade privada de várias VPCs para o Amazon MSK é uma opção de conectividade que permite conectar clientes Apache Kafka hospedados em diferentes nuvens privadas virtuais (VPCs) e AWS contas a um cluster MSK.

O Amazon MSK simplifica o acesso entre contas com políticas de cluster. Essas políticas permitem que o proprietário do cluster conceda permissões para que outras AWS contas estabeleçam conectividade privada com o cluster MSK.

Benefícios da conectividade privada multi-VPC

A conectividade privada multi-VPC tem várias vantagens em relação a outras soluções de conectividade:

- Ele automatiza o gerenciamento operacional da solução de AWS PrivateLink conectividade.
- Ele permite a sobreposição IPs entre conexões VPCs, eliminando a necessidade de manter tabelas de emparelhamento e roteamento complexas e não sobrepostas IPs associadas a outras soluções de conectividade de VPC.

Você usa uma política de cluster para seu cluster MSK para definir quais AWS contas têm permissões para configurar a conectividade privada entre contas com seu cluster MSK. O administrador de várias contas pode delegar permissões aos perfis ou usuários adequados. Quando usada com a autenticação de cliente do IAM, você também pode usar a política de cluster para definir as permissões do plano de dados do Kafka de modo granular para os clientes conectados.

Requisitos e limitações para conectividade privada multi-VPC

Observe estes requisitos de cluster do MSK para executar a conectividade privada multi-VPC:

- A conectividade privada multi-VPC é compatível apenas com o Apache Kafka 2.7.1 ou superior. Certifique-se de que todos os clientes que você use com o cluster do MSK estejam executando versões do Apache Kafka compatíveis com o cluster.
- A conectividade privada multi-VPC é compatível com os tipos de autenticação IAM, TLS e SASL/SCRAM. Clusters não autenticados não podem usar conectividade privada multi-VPC.
- Se você estiver usando os métodos de controle de acesso SASL/SCRAM ou mTLS, deverá configurar o Apache ACLs Kafka para seu cluster. Primeiro, defina o Apache Kafka ACLs para seu cluster. Em seguida, atualize a configuração do cluster para que a propriedade `allow.everyone.if.no.acl.found` seja definida como falsa para o cluster. Para obter informações sobre como atualizar a configuração de um cluster, consulte the section called "Operações de configuração do broker". Se você estiver usando o controle de acesso do IAM e quiser aplicar políticas de autorização ou atualizar suas políticas de autorização, consulte the section called "Controle de acesso do IAM". Para obter informações sobre o Apache Kafka ACLs, consulte. the section called "Apache Kafka ACLs"
- A conectividade privada multi-VPC não é compatível com o tipo de instância t3.small.
- A conectividade privada de várias VPCs não é suportada em todas AWS as regiões, somente em AWS contas dentro da mesma região.
- Para configurar a conectividade privada de várias VPCs, você deve ter o mesmo número de sub-redes de cliente que as sub-redes de cluster. Você também deve garantir que as zonas de disponibilidade IDs sejam as mesmas para a sub-rede do cliente e a sub-rede do cluster.
- O Amazon MSK não é compatível com conectividade privada multi-VPC com os nós do Zookeeper.

Conceitos básicos sobre como usar a conectividade privada de várias VPCs

Tópicos

- Etapa 1: no cluster do MSK na conta A, ativar a conectividade multi-VPC para o esquema de autenticação do IAM no cluster
- Etapa 2: anexar uma política de cluster ao cluster do MSK
- Etapa 3: ações de usuários entre contas para configurar conexões de VPC gerenciadas pelo cliente

Este tutorial usa um caso de uso comum como exemplo de como você pode usar a conectividade de várias VPCs para conectar de forma privada um cliente Apache Kafka a um cluster MSK de

dentro, AWS mas fora da VPC do cluster. Esse processo exige que o usuário entre contas crie uma conexão e uma configuração de VPC gerenciada pelo MSK para cada cliente, incluindo as permissões de cliente necessárias. O processo também exige que o proprietário do cluster MSK habilite a PrivateLink conectividade no cluster MSK e selecione esquemas de autenticação para controlar o acesso ao cluster.

Em diferentes partes deste tutorial, escolhemos as opções aplicáveis a esse exemplo. Isso não significa que estas são as únicas opções disponíveis para configurar um cluster do MSK ou instâncias de cliente.

A configuração de rede para esse caso de uso é a seguinte:

- Um usuário com várias contas (cliente Kafka) e um cluster do MSK estão na mesma rede/região da AWS , mas em contas diferentes:
  - Cluster do MSK na conta A
  - Cliente Kafka na conta B
- O usuário entre contas se conectará de modo privado ao cluster do MSK usando o esquema de autenticação do IAM.

Este tutorial pressupõe que há um cluster do MSK provisionado criado com o Apache Kafka versão 2.7.1 ou superior. O cluster do MSK deve estar em um estado ACTIVE antes de iniciar o processo de configuração. Para evitar possíveis perdas de dados ou tempo de inatividade, os clientes que usarão uma conexão privada multi-VPC para se conectar ao cluster devem usar versões do Apache Kafka compatíveis com o cluster.

O diagrama a seguir ilustra a arquitetura da conectividade multi-VPC do Amazon MSK conectada a um cliente em uma conta diferente. AWS

Etapa 1: no cluster do MSK na conta A, ativar a conectividade multi-VPC para o esquema de autenticação do IAM no cluster

O proprietário do cluster do MSK precisa fazer as configurações no cluster do MSK após a criação do cluster e em um estado ACTIVE.

O proprietário do cluster ativa a conectividade privada multi-VPC no cluster ACTIVE para qualquer esquema de autenticação que estará ativo no cluster. Isso pode ser feito usando a UpdateSecurity API ou o console MSK. Os esquemas de autenticação do IAM, SASL/SCRAM e TLS são compatíveis

com conectividade privada multi-VPC. A conectividade privada multi-VPC não pode ser habilitada para clusters não autenticados.

Para esse caso de uso, você configurará o cluster para usar o esquema de autenticação do IAM.

> **Note**
>
> Se você estiver configurando seu cluster MSK para usar o esquema de SASL/SCRAM autenticação, a propriedade "" do Apache Kafka ACLs é obrigatória. `allow.everyone.if.no.acl.found=false` Veja Apache ACLs Kafka.

Quando você atualiza as configurações de conectividade privada multi-VPC, o Amazon MSK inicia uma reinicialização contínua dos nós do agente que atualiza as configurações do agente. A conclusão dessa operação pode levar até 30 minutos ou mais. Você não pode fazer outras atualizações no cluster enquanto a conectividade estiver sendo atualizada.

Ativar o recurso multi-VPC para esquemas de autenticação selecionados no cluster na conta A usando o console

1. Abra o console do Amazon MSK em https://console.aws.amazon.com/msk/para a conta em que o cluster está localizado.
2. No painel de navegação, em Clusters do MSK, escolha Clusters para exibir a lista de clusters na conta.
3. Selecione o cluster a ser configurado para conectividade privada multi-VPC. O cluster deve estar em um estado ACTIVE.
4. Selecione a guia Propriedades do cluster e acesse as configurações de Rede.
5. Selecione o menu suspenso Editar e selecione Ativar conectividade multi-VPC.
6. Selecione um ou mais tipos de autenticação que você deseja ativar para esse cluster. Para esse caso de uso, selecione a autenticação baseada em perfil do IAM.
7. Selecione Salvar alterações.

Example - UpdateConnectivity API que ativa esquemas de autenticação de conectividade privada de várias VPCs em um cluster

Como alternativa ao console MSK, você pode usar a UpdateConnectivity API para ativar a conectividade privada de várias VPCs e configurar esquemas de autenticação em um cluster ATIVO. O exemplo a seguir mostra o esquema de autenticação do IAM ativado para o cluster.

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

O Amazon MSK cria a infraestrutura de rede necessária para conectividade privada. O Amazon MSK também cria um novo conjunto de endpoints do agente de bootstrap para cada tipo de autenticação que requer conectividade privada. Observe que o esquema de autenticação em texto simples não oferece suporte à conectividade privada multi-VPC.

Etapa 2: anexar uma política de cluster ao cluster do MSK

O proprietário do cluster pode anexar uma política de cluster (também conhecida como política baseada em recurso) ao cluster do MSK no qual você ativará a conectividade privada multi-VPC. A política de cluster permite que os clientes acessem o cluster usando outra conta. Antes de editar a política de cluster, você precisa dos IDs de conta para as contas que devem ter permissão para acessar o cluster do MSK. Consulte Como o Amazon MSK funciona com o IAM.

O proprietário do cluster deve anexar uma política de cluster ao cluster do MSK que autorize o usuário entre contas na conta B a obter agentes de bootstrap para o cluster e a autorizar as seguintes ações no cluster do MSK na conta A:

- CreateVpcConnection
- GetBootstrapBrokers

- DescribeCluster
- DescribeClusterV2

Example

Para referência, veja a seguir um exemplo do JSON para uma política básica de cluster, semelhante à política padrão apresentada no editor de políticas do IAM do console do MSK. A política a seguir concede permissões para acesso em nível de cluster, tópico e grupo.

JSON

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

```
      },
      "Action": "kafka-cluster:*",
      "Resource": "arn:aws:kafka:us-east-1:111122223333:group/testing/*"
    }
  ]
}
```

Anexar uma política de cluster ao cluster do MSK

1. No console do Amazon MSK, em Clusters do MSK, escolha Clusters.
2. Role para baixo até Configurações de segurança e selecione Editar política de cluster.
3. No console, na tela Editar política de cluster, selecione Política básica para conectividade multi-VPC.
4. No campo ID da conta, insira o ID da conta para cada conta que deve ter permissão para acessar esse cluster. Conforme você digita o ID, ele é copiado automaticamente para a sintaxe JSON da política exibida. Em nosso exemplo de política de cluster, o ID da conta é `123456789012`.
5. Selecione Salvar alterações.

Para obter informações sobre a política de cluster APIs, consulte as políticas baseadas em recursos do Amazon MSK.

Etapa 3: ações de usuários entre contas para configurar conexões de VPC gerenciadas pelo cliente

Para configurar a conectividade privada multi-VPC entre um cliente em uma conta diferente do cluster do MSK, o usuário entre contas cria uma conexão VPC gerenciada para o cliente. É possível conectar vários clientes ao cluster do MSK repetindo esse procedimento. Para fins desse caso de uso, você configurará apenas um cliente.

Os clientes podem usar os esquemas de autenticação compatíveis IAM, SASL/SCRAM ou TLS. Cada conexão de VPC gerenciada só pode ter um esquema de autenticação associado a ela. O esquema de autenticação do cliente deve ser configurado no cluster do MSK ao qual o cliente se conectará.

Para esse caso de uso, configure o esquema de autenticação do cliente para que o cliente na conta B use o esquema de autenticação do IAM.

Pré-requisitos

Esse processo requer os seguintes itens:

- A política de cluster criada anteriormente que concede ao cliente na conta B permissão para realizar ações no cluster do MSK na conta A.
- Uma política de identidade anexada ao cliente na conta B que concede permissões para as ações `kafka:CreateVpcConnection`, `ec2:CreateTags`, `ec2:CreateVPCEndpoint` e `ec2:DescribeVpcAttribute`.

Example

Para referência, este é um exemplo do JSON para uma política básica de identidade de cliente.

JSON

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

Para criar uma conexão de VPC gerenciada para um cliente na conta B

1. Do administrador do cluster, obtenha o ARN do cluster do MSK na conta A ao qual você deseja que o cliente na conta B se conecte. Anote o ARN do cluster para usar posteriormente.
2. No console do MSK da conta B do cliente, escolha Conexões VPC gerenciadas e, em seguida, escolha Criar conexão.

3. No painel Configurações de conexão, cole o ARN do cluster no campo de texto ARN do cluster e escolha Verificar.
4. Selecione o Tipo de autenticação para o cliente na conta B. Para esse caso de uso, escolha IAM ao criar a conexão VPC do cliente.
5. Escolha a VPC para o cliente.
6. Escolha pelo menos duas zonas de disponibilidade e sub-redes associadas. Você pode obter a zona IDs de disponibilidade nos detalhes do cluster do AWS Management Console ou usando a DescribeClusterAPI ou o comando da AWS CLI describe-cluster. A zona IDs que você especifica para a sub-rede do cliente deve corresponder às da sub-rede do cluster. Se os valores de uma sub-rede estiverem ausentes, primeiro crie uma sub-rede com o mesmo ID de zona do seu cluster do MSK.
7. Escolha um Grupo de segurança para essa conexão VPC. Você pode usar o grupo de segurança padrão. Para mais informações sobre a configuração de grupos de segurança, consulte Controlar o tráfego para recursos usando grupos de segurança.
8. Selecione Criar conexão.
9. Para obter a lista das novas strings de agente de bootstrap no console do MSK do usuário entre contas (Detalhes do cluster > Conexão VPC gerenciada), consulte as strings de agente de bootstrap exibidas em “Cadeia de conexão do cluster”. Na Conta B do cliente, a lista de corretores de bootstrap pode ser visualizada chamando a GetBootstrapBrokersAPI ou visualizando a lista de corretores de bootstrap nos detalhes do cluster do console.
10. Atualize os grupos de segurança associados às conexões de VPC da seguinte forma:

    a. Defina regras de entrada para a PrivateLink VPC para permitir todo o tráfego do intervalo de IP da rede da Conta B.

    b. [Opcional] Defina as regras de conectividade de saída para o cluster do MSK. Escolha o grupo de segurança no console da VPC, Editar regras de saída e adicione uma regra para o Tráfego TCP personalizado para os intervalos de portas 14001-14100. O balanceador de carga de rede multi-VPC está escutando nos intervalos de portas 14001-14100. Consulte Network Load Balancers.
11. Configure o cliente na conta B para usar os novos agentes de bootstrap para conectividade privada multi-VPC para se conectar ao cluster do MSK na conta A. Consulte Produzir e consumir dados.

Após a conclusão da autorização, o Amazon MSK criará uma conexão VPC gerenciada para cada VPC e esquema de autenticação especificados. O grupo de segurança escolhido será associado a

cada conexão. Essa conexão VPC gerenciada é configurada pelo Amazon MSK para se conectar de maneira privada aos agentes. Você pode usar o novo conjunto de agentes de bootstrap para se conectar de maneira privada ao cluster do Amazon MSK.

Atualizar os esquemas de autorização em um cluster

A conectividade privada de várias VPCs oferece suporte a vários esquemas de autorização: conectividade SASL/SCRAM, IAM, and TLS. The cluster owner can turn on/off privada para um ou mais esquemas de autenticação. O cluster precisa estar no estado ACTIVE para realizar essa ação.

Para ativar um esquema de autenticação usando o console do Amazon MSK

1. Abra o console do Amazon MSK em AWS Management Console para o cluster que deseja editar.
2. No painel de navegação, em Clusters do MSK, escolha Clusters para exibir a lista de clusters na conta.
3. Selecione o cluster que deseja editar. O cluster deve estar em um estado ACTIVE.
4. Selecione a guia Propriedades do cluster e acesse Configurações de rede.
5. Selecione o menu suspenso Editar e selecione Ativar conectividade multi-VPC para ativar o novo esquema de autorização.
6. Selecione um ou mais tipos de autenticação que você deseja ativar para esse cluster.
7. Selecione Ativar seleção.

Ao ativar um novo esquema de autenticação, você também deverá criar novas conexões VPC gerenciadas para o novo esquema de autenticação e atualizar seus clientes para usar os agentes de bootstrap específicos do novo esquema de autenticação.

Para desativar um esquema de autenticação usando o console do Amazon MSK

> **Note**
>
> Quando você desativa a conectividade privada multi-VPC para esquemas de autenticação, toda a infraestrutura relacionada à conectividade é excluída, incluindo as conexões VPC gerenciadas.

Quando você desativa a conectividade privada multi-VPC para esquemas de autenticação, as conexões VPC existentes no lado do cliente mudam para INACTIVE, e a infraestrutura do Privatelink no lado do cluster é removida, incluindo as conexões VPC gerenciadas. O usuário de várias contas só pode excluir a conexão VPC inativa. Se a conectividade privada for ativada novamente no cluster, o usuário entre contas precisará criar uma nova conexão com o cluster.

1. Abra o console do Amazon MSK em AWS Management Console.
2. No painel de navegação, em Clusters do MSK, escolha Clusters para exibir a lista de clusters na conta.
3. Selecione o cluster que deseja editar. O cluster deve estar em um estado ACTIVE.
4. Selecione a guia Propriedades do cluster e acesse Configurações de rede.
5. Selecione o menu suspenso Editar e selecione Desativar conectividade multi-VPC (para desativar um esquema de autorização).
6. Selecione um ou mais tipos de autenticação que você deseja desativar para esse cluster.
7. Selecione Desativar seleção.

Example Para ativar on/off um esquema de autenticação com a API

Como alternativa ao console MSK, você pode usar a UpdateConnectivity API para ativar a conectividade privada de várias VPCs e configurar esquemas de autenticação em um cluster ATIVO. O exemplo a seguir mostra SASL/SCRAM os esquemas de autenticação do IAM ativados para o cluster.

Ao ativar um novo esquema de autenticação, você também deverá criar novas conexões VPC gerenciadas para o novo esquema de autenticação e atualizar seus clientes para usar os agentes de bootstrap específicos do novo esquema de autenticação.

Quando você desativa a conectividade privada multi-VPC para esquemas de autenticação, as conexões VPC existentes no lado do cliente mudam para INACTIVE, e a infraestrutura do Privatelink no lado do cluster é removida, incluindo as conexões VPC gerenciadas. O usuário de várias contas só pode excluir a conexão VPC inativa. Se a conectividade privada for ativada novamente no cluster, o usuário entre contas precisará criar uma nova conexão com o cluster.

```
Request:
{
  "currentVersion": "string",
  "connnectivityInfo": {
```

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

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

Rejeitar uma conexão VPC gerenciada com um cluster do Amazon MSK

Você pode rejeitar uma conexão VPC do cliente no console do Amazon MSK na conta de administrador do cluster. Para ser rejeitada, a conexão VPC do cliente deve estar no estado AVAILABLE. Talvez você queira rejeitar uma conexão VPC gerenciada de um cliente que não esteja mais autorizado a se conectar ao seu cluster. Para evitar que novas conexões VPC gerenciadas se conectem a um cliente, negue o acesso ao cliente na política de cluster. Uma conexão rejeitada ainda gera custos até ser excluída pelo proprietário da conexão. Consulte Excluir uma conexão VPC gerenciada com um cluster do Amazon MSK.

Para rejeitar uma conexão VPC do cliente usando o console do MSK

1. Abra o console do Amazon MSK em AWS Management Console.
2. No painel de navegação, selecione Clusters e localize a lista Configurações de rede > Conexões VPC do cliente.

3. Selecione a conexão que deseja rejeitar e selecione Rejeitar conexão VPC do cliente.
4. Confirme que deseja rejeitar a conexão VPC do cliente selecionada.

Para rejeitar uma conexão VPC gerenciada usando a API, use a API `RejectClientVpcConnection`.

Excluir uma conexão VPC gerenciada com um cluster do Amazon MSK

O usuário entre contas pode excluir uma conexão VPC gerenciada para um cluster do MSK no console da conta do cliente. Como o usuário proprietário do cluster não é proprietário da conexão VPC gerenciada, a conexão não pode ser excluída na conta de administrador do cluster. Após a exclusão de uma conexão VPC, ela não terá mais custos.

Para excluir uma conexão VPC gerenciada usando o console do MSK

1. Na conta do cliente, abra o console do Amazon MSK em AWS Management Console.
2. No painel de navegação, selecione Conexões VPC gerenciadas.
3. Na lista de conexões, selecione a conexão que deseja excluir.
4. Confirme que deseja excluir a conexão VPC.

Para excluir uma conexão VPC gerenciada usando a API, use a API `DeleteVpcConnection`.

Permissões para conectividade privada multi-VPC

Esta seção resume as permissões necessárias para clientes e clusters que usam o recurso de conectividade privada multi-VPC. A conectividade privada multi-VPC exige que o administrador do cliente crie permissões em cada cliente que terá uma conexão VPC gerenciada com o cluster do MSK. Também exige que o administrador do cluster MSK habilite a PrivateLink conectividade no cluster MSK e selecione esquemas de autenticação para controlar o acesso ao cluster.

Tipo de autenticação de cluster e permissões de acesso a tópicos

Ative o recurso de conectividade privada multi-VPC para esquemas de autenticação habilitados para seu cluster do MSK. Consulte Requisitos e limitações para conectividade privada multi-VPC. Se você estiver configurando seu cluster MSK para usar o esquema de SASL/SCRAM autenticação, a propriedade Apache ACLs Kafka é obrigatória. `allow.everyone.if.no.acl.found=false` Após definir as Apache Kafka ACLs para seu cluster, atualize a configuração do cluster para

que a propriedade `allow.everyone.if.no.acl.found` seja falsa para o cluster. Para obter informações sobre como atualizar a configuração de um cluster, consulte Operações de configuração do broker.

Permissões de política de cluster entre contas

Se um cliente Kafka estiver em uma AWS conta diferente do cluster MSK, anexe uma política baseada em cluster ao cluster MSK que autorize o usuário raiz do cliente a conectividade entre contas. Você pode editar a política de cluster de várias VPCs usando o editor de políticas do IAM no console MSK (configurações de segurança do cluster > Editar política do cluster) ou usar o seguinte APIs para gerenciar a política do cluster:

PutClusterPolicy

Anexa a política de cluster ao cluster. Você pode usar essa API para criar ou atualizar a política de cluster do MSK especificada. Se você estiver atualizando a política, o campo currentVersion será obrigatório na carga da solicitação.

GetClusterPolicy

Recupera o texto JSON do documento de política de cluster anexado ao cluster.

DeleteClusterPolicy

Exclui a política de cluster.

Veja a seguir um exemplo do JSON para uma política básica de cluster, semelhante à política padrão apresentada no editor de políticas do IAM do console do MSK. A política a seguir concede permissões para acesso em nível de cluster, tópico e grupo.

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/*"
        ]
    }]
}
```

Permissões de cliente para conectividade privada multi-VPC com um cluster do MSK

Para configurar a conectividade privada multi-VPC entre um cliente Kafka e um cluster do MSK, o cliente precisa ter uma política de identidade anexada que conceda permissões para as ações `kafka:CreateVpcConnection`, `ec2:CreateTags` e `ec2:CreateVPCEndpoint` no cliente. Para referência, este é um exemplo do JSON para uma política básica de identidade de cliente.

JSON

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

Informações de porta

Use os seguintes números de porta para que o Amazon MSK possa se comunicar com máquinas clientes:

- Para se comunicar com agentes em texto simples, os agentes usam a porta 9092.
- Para se comunicar com corretores com criptografia TLS, use a porta 9094 para acesso interno AWS e a porta 9194 para acesso público.
- Para se comunicar com corretores com SASL/SCRAM, use a porta 9096 para acesso interno AWS e a porta 9196 para acesso público.
- Para se comunicar com corretores em um cluster configurado para usothe section called “Controle de acesso do IAM”, use a porta 9098 para acesso interno AWS e a porta 9198 para acesso público.
- Para se comunicar com o Apache ZooKeeper usando a criptografia TLS, use a porta 2182. ZooKeeper Os nós Apache usam a porta 2181 por padrão.

Obter os agentes de bootstrap para um cluster do Amazon MSK

O termo agentes de bootstrap refere-se a uma lista de agentes que um cliente Apache Kafka pode usar para se conectar ao cluster do Amazon MSK. Essa lista pode não incluir todos os agentes no cluster. Você pode obter corretores de bootstrap usando a API AWS Management Console AWS CLI, ou Amazon MSK.

Tópicos

- Obtenha os corretores de bootstrap usando o AWS Management Console
- Obtenha os corretores de bootstrap usando o AWS CLI
- Obter os agentes de bootstrap usando a API

Obtenha os corretores de bootstrap usando o AWS Management Console

Esse processo descreve como obter corretores de bootstrap para um cluster usando o. AWS Management Console O termo agentes de bootstrap se refere a uma lista de agentes que um cliente Apache Kafka pode usar como ponto de partida para se conectar ao cluster. Essa lista não inclui necessariamente todos os agentes em um cluster.

1. Faça login no AWS Management Console e abra o console do Amazon MSK em https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. A tabela mostra todos os clusters da região atual nesta conta. Escolha o nome de um cluster para visualizar sua descrição.
3. Na página Resumo do cluster, escolha Exibir informações do cliente. Isso mostra os corretores de bootstrap, bem como a string de conexão do Apache ZooKeeper .

Obtenha os corretores de bootstrap usando o AWS CLI

Execute o comando a seguir, substituindo *ClusterArn* pelo nome do recurso da Amazon (ARN) que você obteve quando criou o cluster. Se você não tiver o ARN do cluster, poderá encontrá-lo listando todos os clusters. Para obter mais informações, consulte the section called “Listar clusters”.

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

Para um cluster do MSK que use o the section called “Controle de acesso do IAM”, a saída desse comando é semelhante ao seguinte exemplo de JSON.

```
{
    "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"
}
```

O exemplo a seguir mostra os agentes de bootstrap de um cluster com acesso público ativado. Use o `BootstrapBrokerStringPublicSaslIam` para acesso público e a `BootstrapBrokerStringSaslIam` string para acesso interno AWS.

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

A string de agentes de bootstrap deve conter três agentes de todas as zonas de disponibilidade nas quais seu cluster do MSK esteja implantado (a menos que haja apenas dois agentes disponíveis).

Obter os agentes de bootstrap usando a API

Para fazer com que os corretores de bootstrap usem a API, consulte. GetBootstrapBrokers

Monitore um cluster provisionado do Amazon MSK

Há várias maneiras pelas quais o Amazon MSK ajuda você a monitorar o status do seu cluster provisionado do Amazon MSK.

- O Amazon MSK reúne métricas do Apache Kafka e as envia para a Amazon, CloudWatch onde você pode visualizá-las. Para obter mais informações sobre as métricas do Apache Kafka, incluindo as que surgem com o Amazon MSK, consulte Monitoramento na documentação do Apache Kafka.
- Também é possível monitorar o cluster do MSK com o Prometheus, uma aplicação de código aberto para monitoramento. Para obter informações sobre o Prometheus, consulte Visão geral na documentação do Prometheus. Para saber como monitorar seu cluster provisionado pelo MSK com o Prometheus, consulte. the section called “Monitorare com o Prometheus”
- (Somente corretores padrão) O Amazon MSK ajuda você a monitorar sua capacidade de armazenamento em disco enviando automaticamente alertas de capacidade de armazenamento quando um cluster provisionado está prestes a atingir seu limite de capacidade de armazenamento. Os alertas também fornecem recomendações sobre as melhores etapas a serem seguidas para resolver os problemas detectados. Isso ajuda você a identificar e resolver rapidamente os problemas de capacidade de disco antes que eles se tornem críticos. O Amazon MSK envia automaticamente esses alertas para o console do Amazon MSK, para a AWS Health Dashboard Amazon EventBridge e para os contatos de e-mail da sua AWS conta. Para obter mais informações sobre alertas de capacidade de armazenamento, consulte Usar alertas de capacidade de armazenamento do Amazon MSK.

Tópicos

- Veja as métricas do Amazon MSK usando CloudWatch
- Métricas do Amazon MSK para monitorar corretores padrão com CloudWatch
- Métricas do Amazon MSK para monitorar corretores Express com CloudWatch
- Monitore um cluster provisionado pelo MSK com o Prometheus

- Monitorar atrasos do consumidor
- Usar alertas de capacidade de armazenamento do Amazon MSK

Veja as métricas do Amazon MSK usando CloudWatch

Você pode monitorar as métricas do Amazon MSK usando o CloudWatch console, a linha de comando ou a CloudWatch API. Os procedimentos a seguir mostram como acessar as métricas usando os seguintes métodos:

Para acessar métricas usando o CloudWatch console

Faça login no AWS Management Console e abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.

1. No painel de navegação, selecione Métricas.
2. Escolha a guia Todas as métricas e escolha AWS/Kafka.
3. Para visualizar métricas em nível de tópico, escolha Topic, Broker ID, Cluster Name (Tópico, ID do agente, nome do cluster); para métricas em nível de agente, escolha Broker ID, Cluster Name (ID do agente, nome do cluster) e, para métricas em nível de cluster, escolha Cluster Name (Nome do cluster).
4. (Opcional) No painel gráfico, selecione uma estatística e um período de tempo e, em seguida, crie um CloudWatch alarme usando essas configurações.

Para acessar métricas usando o AWS CLI

Use as métricas e get-metric-statisticsos comandos da lista.

Para acessar métricas usando a CloudWatch CLI

Use os comandos mon-list-metrics e mon-get-stats.

Para acessar métricas usando a CloudWatch API

Use as operações ListMetrics e GetMetricStatistics.

Métricas do Amazon MSK para monitorar corretores padrão com CloudWatch

O Amazon MSK se integra à Amazon CloudWatch para que você possa coletar, visualizar e analisar CloudWatch métricas para seus corretores MSK Standard. As métricas que você configura para seus clusters provisionados pelo MSK são coletadas e enviadas automaticamente em

intervalos de 1 CloudWatch minuto. Você pode definir o nível de monitoramento de um cluster provisionado pelo MSK como um dos seguintes:`DEFAULT`,,`PER_BROKER`, `PER_TOPIC_PER_BROKER` ou. `PER_TOPIC_PER_PARTITION` As tabelas nas seções a seguir mostram todas as métricas disponíveis em cada nível de monitoramento.

> **Note**
>
> Os nomes de algumas métricas do Amazon MSK para CloudWatch monitoramento foram alterados na versão 3.6.0 e superior. Use os novos nomes para monitorar essas métricas. Para métricas com nomes alterados, a tabela abaixo mostra o nome usado nas versões 3.6.0 e posteriores, seguido pelo nome na versão 2.8.2.tiered.

As métricas no nível `DEFAULT` são gratuitas. Os preços de outras métricas estão descritos na página de CloudWatchpreços da Amazon.

Monitoramento no nível **`DEFAULT`**

As métricas descritas na tabela a seguir estão disponíveis no nível de monitoramento `DEFAULT`. Elas são gratuitas.

Name	Quando visível	Dimensõ	Descrição
`ActiveController Count`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster	Somente um controlador por cluster deve estar ativo em qualquer momento.
`BurstBalance`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O saldo restante dos créditos de intermitência de entrada/saída para volumes do EBS no cluster. Use-o para investigar a latência ou a diminuição do throughput. `BurstBalance` não é relatado para volumes do EBS quando o desempenho de linha de base de um volume for maior que o desempenho máximo de intermitência. Para obter mais informações, consulte Créditos

Name	Quando visível	Dimensõ	Descrição
			de E/S e desempenho de intermitência.
`BytesInPerSec`	Depois de criar um tópico.	Nome do cluster, ID do agente, tópico	O número de bytes por segundo recebidos dos clientes. Essa métrica está disponível por agente e também por tópico.
`BytesOutPerSec`	Depois de criar um tópico.	Nome do cluster, ID do agente, tópico	O número de bytes por segundo enviados aos clientes. Essa métrica está disponível por agente e também por tópico.
`ClientConnectionCount`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente, autenticação de cliente	O número de conexões de cliente autenticadas e ativas.
`ConnectionCount`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de conexões ativas autenticadas, não autenticadas e entre agentes.

Name	Quando visível	Dimensõ	Descrição
`CPUCredit Balance`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de créditos ganhos de CPU que um agente acumulou desde que foi iniciado. Os créditos são acumulados no saldo de créditos após terem sido ganhos e são removidos do saldo de créditos quando são gastos. A falta de saldo de créditos de CPU pode afetar negativamente o desempenho do cluster. Você pode adotar medidas para reduzir a carga da CPU. Por exemplo, você pode reduzir o número de solicitações de clientes ou atualizar o tipo de agente para um tipo de agente M5.
`CpuIdle`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	A porcentagem de tempo ocioso da CPU.
`CpuIoWait`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O percentual de tempo ocioso da CPU durante uma operação de disco pendente.
`CpuSystem`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	A porcentagem de CPU no espaço do kernel.

Name	Quando visível	Dimensõ	Descrição
`CpuUser`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	A porcentagem de CPU no espaço do usuário.
`GlobalPartitionCount`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster	O número de partições em todos os tópicos no cluster, excluindo réplicas. Como `GlobalPartitionCount` não inclui réplicas, a soma dos `PartitionCount` valores pode ser maior do que GlobalPartitionCount se o fator de replicação de um tópico for maior que 1.
`GlobalTopicCount`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster	Número total de tópicos em todos os agentes no cluster.
`EstimatedMaxTimeLag`	Depois que o grupo de consumidores consome de um tópico.	Nome do cluster, grupo de consumidores, tópico	Estimativa de tempo (em segundos) para drenar `MaxOffsetLag` .
`KafkaAppLogsDiskUsed`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	A porcentagem de espaço em disco usada para logs de aplicativos.

Name	Quando visível	Dimensõ	Descrição
`KafkaDataLogsDiskUsed` (dimensão `Cluster Name, Broker ID`)	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	A porcentagem de espaço em disco usada para logs de dados.
`LeaderCount`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número total de líderes de partições por agente, sem incluir réplicas.
`MaxOffsetLag`	Depois que o grupo de consumidores consome de um tópico.	Nome do cluster, grupo de consumi res, tópico	O atraso máximo de deslocame nto entre todas as partições em um tópico.
`MemoryBuffered`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O tamanho, em bytes, da memória armazenada em buffer para o agente.
`MemoryCached`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O tamanho, em bytes, da memória armazenada em cache para o agente.

Name	Quando visível	Dimensõ	Descrição
`MemoryFree`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O tamanho, em bytes, de memória que é gratuita e disponível para o agente.
`HeapMemoryAfterGC`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O percentual da memória total da pilha em uso após a coleta de resíduos.
`MemoryUsed`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O tamanho, em bytes, de memória que está em uso pelo agente.
`MessagesInPerSec`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de mensagens recebidas por segundo do agente.
`NetworkRxDropped`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de pacotes de recebimento descartados.
`NetworkRxErrors`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de erros de recepção da rede para o agente.

Name	Quando visível	Dimensõ	Descrição
`NetworkRx Packets`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de pacotes recebidos pelo agente.
`NetworkTx Dropped`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de pacotes de transmissão descartados.
`NetworkTx Errors`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de erros de transmissão da rede para o agente.
`NetworkTx Packets`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de pacotes transmitidos pelo agente.
`OfflinePa rtitionsCount`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster	Número total de partições que estão offline no cluster.
`PartitionCount`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número total de partições de tópico por agente, incluindo réplicas.

Name	Quando visível	Dimensõ	Descrição
`ProduceTotalTimeMsMean`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O tempo médio de produção em milissegundos.
`RequestBytesMean`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número médio de bytes de solicitações do agente.
`RequestTime`	Após a limitação da solicitação ser aplicada.	Nome do cluster, ID do agente	O tempo médio gasto em milissegundos em threads de rede e de E/S do agente para processar solicitações.
`RootDiskUsed`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	A porcentagem do disco raiz usado pelo agente.
`SumOffsetLag`	Depois que o grupo de consumidores consome de um tópico.	Nome do cluster, grupo de consumires, tópico	O atraso de deslocamento agregado para todas as partições em um tópico.

Name	Quando visível	Dimensõ	Descrição
SwapFree	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O tamanho, em bytes, de memória de swap que está disponível para o agente.
SwapUsed	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O tamanho em bytes de memória de swap que está em uso para o agente.
TrafficShaping	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	Métricas de alto nível que indicam o número de pacotes modelados (descartados ou enfileirados) devido ao excesso de alocações de rede. É possível obter detalhes mais aprofundados com as métricas de PER_BROKER.
UnderMinIsrPartitionCount	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de partições em minIsr do agente.
UnderReplicatedPartitions	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de partições sub-replicadas do agente.

Name	Quando visível	Dimensõ	Descrição
`UserPartitionExists`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	Uma métrica booleana que indica a presença de uma partição de propriedade do usuário em uma corretora. Um valor de 1 indica a presença de partições no corretor.
`ZooKeeperRequestLatencyMsMean`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	Para cluster ZooKeeper baseado. A latência média em milissegundos para ZooKeeper solicitações do Apache do broker.
`ZooKeeperSessionState`	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	Para cluster ZooKeeper baseado. Status da conexão da ZooKeeper sessão do broker, que pode ser um dos seguintes: NOT_CONNECTED: '0.0', ASSOCIATING: '0.1', CONNECTING: '0.5', CONNECTEDREADONLY: '0.8', CONNECTED: '1.0', CLOSED: '5.0', AUTH_FAILED: '10.0'.

Monitoramento no nível **`PER_BROKER`**

Ao definir o nível de monitoramento como `PER_BROKER`, você obtém as métricas descritas na tabela a seguir, além de todas as métricas de nível `DEFAULT`. Você paga pelas métricas na tabela a seguir, enquanto as métricas de nível `DEFAULT` continuam gratuitas. As métricas nesta tabela têm as seguintes dimensões: nome do cluster, ID do agente.

Name	Quando visível	Descrição
`BwInAllowanceExceeded`	Depois que o cluster passa para o estado ACTIVE.	Número de pacotes formados porque a largura de banda agregada de entrada excedeu o máximo para o agente.

Name	Quando visível	Descrição
`BwOutAllowanceExceeded`	Depois que o cluster passa para o estado ACTIVE.	Número de pacotes formados porque a largura de banda agregada de saída excedeu o máximo para o agente.
`ConntrackAllowanceExceeded`	Depois que o cluster passa para o estado ACTIVE.	Número de pacotes formados porque o monitoramento de conexão excedeu o máximo para o agente. O monitoramento de conexão está relacionado a grupos de segurança que monitoram cada conexão estabelecida a fim de garantir que os pacotes de retorno sejam entregues conforme esperado.
`ConnectionCloseRate`	Depois que o cluster passa para o estado ACTIVE.	O número de conexões fechadas por segundo por receptor. Esse número é agregado por receptor e filtrado para os receptores do cliente.
`ConnectionCreationRate`	Depois que o cluster passa para o estado ACTIVE.	O número de novas conexões estabelecidas por segundo por receptor. Esse número é agregado por receptor e filtrado para os receptores do cliente.
`CpuCreditUsage`	Depois que o cluster passa para o estado ACTIVE.	O número de créditos de CPU gastos pelo agente. A falta de saldo de créditos de CPU pode afetar negativamente o desempenho do cluster. Você pode adotar medidas para reduzir a carga da CPU. Por exemplo, você pode reduzir o número de solicitações de clientes ou atualizar o tipo de agente para um tipo de agente M5.

Name	Quando visível	Descrição
`FetchConsumerLocalTimeMsMean`	Depois de haver um produtor/consumidor.	O tempo médio em milissegundos que a solicitação do consumidor é processada no líder.
`FetchConsumerRequestQueueTimeMsMean`	Depois de haver um produtor/consumidor.	O tempo médio em milissegundos que a solicitação do consumidor aguarda na fila de solicitações.
`FetchConsumerResponseQueueTimeMsMean`	Depois de haver um produtor/consumidor.	O tempo médio em milissegundos que a solicitação do consumidor aguarda na fila de resposta.
`FetchConsumerResponseSendTimeMsMean`	Depois de haver um produtor/consumidor.	O tempo médio, em milissegundos, para que o consumidor envie uma resposta.
`FetchConsumerTotalTimeMsMean`	Depois de haver um produtor/consumidor.	O tempo total médio em milissegundos que os consumidores gastam obtendo dados do agente.
`FetchFollowerLocalTimeMsMean`	Depois de haver um produtor/consumidor.	O tempo médio em milissegundos que a solicitação do seguidor é processada no líder.
`FetchFollowerRequestQueueTimeMsMean`	Depois de haver um produtor/consumidor.	O tempo médio em milissegundos que a solicitação de seguidor aguarda na fila de solicitações.
`FetchFollowerResponseQueueTimeMsMean`	Depois de haver um produtor/consumidor.	O tempo médio em milissegundos que a solicitação de seguidor aguarda na fila de resposta.
`FetchFollowerResponseSendTimeMsMean`	Depois de haver um produtor/consumidor.	O tempo médio em milissegundos para o seguidor enviar uma resposta.

Name	Quando visível	Descrição
FetchFollowerTotalTimeMsMean	Depois de haver um produtor/consumidor.	O tempo total médio em milissegundos que os seguidores gastam obtendo e dados do agente.
FetchMessageConversionsPerSec	Depois de criar um tópico.	O número de conversões de mensagens de busca por segundo do agente.
FetchThrottleByteRate	Depois que a limitação da largura de banda é aplicada.	O número de bytes limitados por segundo.
FetchThrottleQueueSize	Depois que a limitação da largura de banda é aplicada.	O número de mensagens na fila de limitação.
FetchThrottleTime	Depois que a limitação da largura de banda é aplicada.	O tempo médio de limitações de busca em milissegundos.
IAMNumberOfConnectionRequests	Depois que o cluster passa para o estado ACTIVE.	O número de solicitações de autenticação do IAM por segundo.
IAMTooManyConnections	Depois que o cluster passa para o estado ACTIVE.	O número de conexões tentadas acima de 100. 0 significa que o número de conexões está dentro do limite. Se >0, o limite do controle de utilização está sendo excedido e você precisa reduzir o número de conexões.
NetworkProcessorAvgIdlePercent	Depois que o cluster passa para o estado ACTIVE.	A porcentagem média do tempo em que os processadores de rede estão ociosos.

Name	Quando visível	Descrição
PpsAllowanceExceeded	Depois que o cluster passa para o estado ACTIVE.	O número de pacotes formados porque o PPS bidirecional excedeu o máximo para o agente.
ProduceLocalTimeMsMean	Depois que o cluster passa para o estado ACTIVE.	O tempo médio em milissegundos que a solicitação leva para ser processada no líder.
ProduceMessageConversionsPerSec	Depois de criar um tópico.	O número de conversões de mensagens de produção por segundo do agente.
ProduceMessageConversionsTimeMsMean	Depois que o cluster passa para o estado ACTIVE.	O tempo médio em milissegundos gasto em conversões de formato de mensagem.
ProduceRequestQueueTimeMsMean	Depois que o cluster passa para o estado ACTIVE.	O tempo médio em milissegundos que as mensagens de solicitação gastam na fila.
ProduceResponseQueueTimeMsMean	Depois que o cluster passa para o estado ACTIVE.	O tempo médio em milissegundos que as mensagens de resposta gastam na fila.
ProduceResponseSendTimeMsMean	Depois que o cluster passa para o estado ACTIVE.	O tempo médio em milissegundos gasto no envio de mensagens de resposta.
ProduceThrottleByteRate	Depois que a limitação da largura de banda é aplicada.	O número de bytes limitados por segundo.
ProduceThrottleQueueSize	Depois que a limitação da largura de banda é aplicada.	O número de mensagens na fila de limitação.

Name	Quando visível	Descrição
`ProduceThrottleTime`	Depois que a limitação da largura de banda é aplicada.	O tempo médio de limitação da produção em milissegundos.
`ProduceTotalTimeMsMean`	Depois que o cluster passa para o estado ACTIVE.	O tempo médio de produção em milissegundos.
`RemoteFetchBytesPerSec (RemoteBytesInPerSec in v2.8.2.tiered)`	Depois de haver um produtor/consumidor.	O número total de bytes transferidos do armazenamento em camadas como resposta às buscas do consumidor. Essa métrica inclui todas as partições de tópicos que contribuem para o tráfego de transferência de dados downstream. Categoria: taxas de tráfego e erro. Essa é uma métrica KIP-405.
`RemoteCopyBytesPerSec (RemoteBytesOutPerSec in v2.8.2.tiered)`	Depois de haver um produtor/consumidor.	O número total de bytes transferidos para o armazenamento em camadas, incluindo dados de segmentos de log, índices e outros arquivos auxiliares. Essa métrica inclui todas as partições de tópicos que contribuem para o tráfego de transferência de dados upstream. Categoria: taxas de tráfego e erro. Essa é uma métrica KIP-405.
`RemoteLogManagerTasksAvgIdlePercent`	Depois que o cluster passa para o estado ACTIVE.	O percentual médio do tempo que o gerenciador remoto de logs ficou ocioso. O gerenciador remoto de logs transfere dados do agente para o armazenamento em camadas. Categoria: atividade interna. Essa é uma métrica KIP-405.

Name	Quando visível	Descrição
`RemoteLogReaderAvgIdlePercent`	Depois que o cluster passa para o estado ACTIVE.	O percentual médio do tempo que o leitor remoto de logs ficou ocioso. O leitor remoto de logs transfere dados do armazenamento remoto para o agente em resposta às buscas do consumidor. Categoria: atividade interna. Essa é uma métrica KIP-405.
`RemoteLogReaderTaskQueueSize`	Depois que o cluster passa para o estado ACTIVE.	O número de tarefas responsáveis por leituras do armazenamento em camadas que estão aguardando para serem agendadas. Categoria: atividade interna. Essa é uma métrica KIP-405.
`RemoteFetchErrorsPerSec (RemoteReadErrorPerSec in v2.8.2.tiered)`	Depois que o cluster passa para o estado ACTIVE.	A taxa total de erros em resposta às solicitações de leitura que o agente especificado enviou ao armazenamento em camadas para recuperar dados em resposta às buscas do consumidor. Essa métrica inclui todas as partições de tópicos que contribuem para o tráfego de transferência de dados downstream. Categoria: taxas de tráfego e erro. Essa é uma métrica KIP-405.

Name	Quando visível	Descrição
`RemoteFetchRequestsPerSec (RemoteReadRequestsPerSec in v2.8.2.tiered)`	Depois que o cluster passa para o estado ACTIVE.	O número total de solicitações de leitura que o agente especificado enviou ao armazenamento em camadas para recuperar dados em resposta às buscas do consumidor. Essa métrica inclui todas as partições de tópicos que contribuem para o tráfego de transferência de dados downstream. Categoria: taxas de tráfego e erro. Essa é uma métrica KIP-405.
`RemoteCopyErrorsPerSec (RemoteWriteErrorPerSec in v2.8.2.tiered)`	Depois que o cluster passa para o estado ACTIVE.	A taxa total de erros em resposta às solicitações de gravação que o agente especificado enviou ao armazenamento em camadas para transferir dados upstream. Essa métrica inclui todas as partições de tópicos que contribuem para o tráfego de transferência de dados upstream. Categoria: taxas de tráfego e erro. Essa é uma métrica KIP-405.
`RemoteLogSizeBytes`	Depois que o cluster passa para o estado ACTIVE.	O número de bytes armazenados na camada remota. Essa métrica está disponível para clusters de armazenamento hierárquico do Apache Kafka versão 3.7.x no Amazon MSK.
`ReplicationBytesInPerSec`	Depois de criar um tópico.	O número de bytes por segundo recebidos dos outros agentes.
`ReplicationBytesOutPerSec`	Depois de criar um tópico.	O número de bytes por segundo enviados para outros agentes.

Name	Quando visível	Descrição
`RequestExemptFromThrottleTime`	Após a limitação da solicitação ser aplicada.	O tempo médio gasto em milissegundos em threads de rede e de E/S do agente para processar solicitações isentas de limitação.
`RequestHandlerAvgIdlePercent`	Depois que o cluster passa para o estado ACTIVE.	A porcentagem média do tempo em que os threads do manipulador de solicitações estão ociosos.
`RequestThrottleQueueSize`	Após a limitação da solicitação ser aplicada.	O número de mensagens na fila de limitação.
`RequestThrottleTime`	Após a limitação da solicitação ser aplicada.	O tempo médio da limitação de solicitações em milissegundos.
`TcpConnections`	Depois que o cluster passa para o estado ACTIVE.	Mostra o número de segmentos TCP de entrada e saída com o sinalizador SYN definido.
`RemoteCopyLagBytes (TotalTierBytesLag in v2.8.2.tiered)`	Depois de criar um tópico.	O número total de bytes dos dados que são elegíveis para classificação hierárquica no agente, mas que ainda não foram transferidos para o armazenamento em camadas. Essas métricas mostram a eficiência da transferência de dados upstream. Conforme o atraso aumenta, a quantidade de dados que não persiste no armazenamento em camadas aumenta. Categoria: atraso de arquivamento. Essa não é uma métrica KIP-405.

Name	Quando visível	Descrição
`TrafficBytes`	Depois que o cluster passa para o estado ACTIVE.	Mostra o tráfego de rede em bytes gerais entre clientes (produtores e consumidores) e agentes. O tráfego entre agentes não é relatado.
`VolumeQueueLength`	Depois que o cluster passa para o estado ACTIVE.	O número de solicitações de operação de leitura e gravação aguardando conclusão em um período especific ado.
`VolumeReadBytes`	Depois que o cluster passa para o estado ACTIVE.	O número de bytes lidos durante um período especificado.
`VolumeReadOps`	Depois que o cluster passa para o estado ACTIVE.	O número de operações de leitura durante um período especificado.
`VolumeTotalReadTime`	Depois que o cluster passa para o estado ACTIVE.	O número total de segundos gastos por todas as operações de leitura que foram concluídas durante um período especificado.
`VolumeTotalWriteTime`	Depois que o cluster passa para o estado ACTIVE.	O número total de segundos gastos por todas as operações de gravação que foram concluídas durante um período especificado.
`VolumeWriteBytes`	Depois que o cluster passa para o estado ACTIVE.	O número de bytes gravados durante um período especificado.
`VolumeWriteOps`	Depois que o cluster passa para o estado ACTIVE.	O número de operações de gravação durante um período especificado.

Monitoramento no nível **PER_TOPIC_PER_BROKER**

Ao definir o nível de monitoramento como PER_TOPIC_PER_BROKER, você obtém as métricas descritas na tabela a seguir, além de todas as métricas dos níveis PER_BROKER e DEFAULT. Somente as métricas de nível DEFAULT são gratuitas. As métricas nesta tabela têm as seguintes dimensões: nome do cluster, ID do agente, tópico.

> **Important**
>
> Para um cluster do Amazon MSK que use o Apache Kafka 2.4.1 ou uma versão mais recente, as métricas na tabela a seguir só aparecerão depois que os valores ficarem diferentes de zero pela primeira vez. Por exemplo, para ver BytesInPerSec, um ou mais produtores devem primeiro enviar dados para o cluster.

Name	Quando visível	Descrição
FetchMessageConversionsPerSec	Depois de criar um tópico.	O número de mensagens obtidas convertidas por segundo.
MessagesInPerSec	Depois de criar um tópico.	O número de mensagens recebidas por segundo.
ProduceMessageConversionsPerSec	Depois de criar um tópico.	O número de conversões por segundo de mensagens produzidas.
RemoteFetchBytesPerSec (RemoteBytesInPerSec in v2.8.2.tiered)	Após criar um tópico e o tópico estiver produzindo/consumindo.	O número de bytes transferidos do armazenamento em camadas em resposta às buscas do consumidor para o tópico e o agente especificados. Essa métrica inclui todas as partições do tópico que contribuem para o tráfego de transferência de dados downstream no agente especificado. Categoria: taxas de tráfego e erro. Essa é uma métrica KIP-405.
RemoteCopyBytesPerSec (RemoteBy	Após criar um tópico e o tópico estiver	O número de bytes transferidos para o armazenamento em camadas, para o tópico e o agente especificados. Essa métrica inclui todas

Name	Quando visível	Descrição
`tesOutPerSec in v2.8.2.tiered)`	produzindo/ consumindo.	as partições do tópico que contribuem para o tráfego de transferência de dados upstream no agente especificado. Categoria: taxas de tráfego e erro. Essa é uma métrica KIP-405.
`RemoteFetchErrorsPerSec (RemoteReadErrorPerSec in v2.8.2.tiered)`	Após criar um tópico e o tópico estiver produzindo/ consumindo.	A taxa de erros em resposta às solicitações de leitura que o agente especificado envia ao armazenamento em camadas para recuperar dados em resposta às buscas do consumidor sobre o tópico especificado. Essa métrica inclui todas as partições do tópico que contribuem para o tráfego de transferência de dados downstream no agente especificado. Categoria: taxas de tráfego e erro. Essa é uma métrica KIP-405.
`RemoteFetchRequestsPerSec (RemoteReadRequestsPerSec in v2.8.2.tiered)`	Após criar um tópico e o tópico estiver produzindo/ consumindo.	O número de solicitações de leitura que o agente especificado envia ao armazenamento em camadas para recuperar dados em resposta às buscas do consumidor sobre o tópico especificado. Essa métrica inclui todas as partições do tópico que contribuem para o tráfego de transferência de dados downstream no agente especificado. Categoria: taxas de tráfego e erro. Essa é uma métrica KIP-405.
`RemoteCopyErrorsPerSec (RemoteWriteErrorPerSec in v2.8.2.tiered)`	Após criar um tópico e o tópico estiver produzindo/ consumindo.	A taxa de erros em resposta às solicitações de gravação que o agente especificado envia ao armazenamento em camadas para transferir dados upstream. Essa métrica inclui todas as partições do tópico que contribuem para o tráfego de transferência de dados upstream no agente especificado. Categoria: taxas de tráfego e erro. Essa é uma métrica KIP-405.

Name	Quando visível	Descrição
`RemoteLogSizeBytes`	Depois de criar um tópico.	O número de bytes armazenados na camada remota. Essa métrica está disponível para clusters de armazenamento hierárquico do Apache Kafka versão 3.7.x no Amazon MSK.

Monitoramento no nível **`PER_TOPIC_PER_PARTITION`**

Ao definir o nível de monitoramento como `PER_TOPIC_PER_PARTITION`, você obtém as métricas descritas na tabela a seguir, além de todas as métricas dos níveis `PER_TOPIC_PER_BROKER`, `PER_BROKER` e DEFAULT. Somente as métricas de nível `DEFAULT` são gratuitas. As métricas nesta tabela têm as seguintes dimensões: grupo de consumidores, tópico, partição.

Name	Quando visível	Descrição
`EstimatedTimeLag`	Depois que o grupo de consumidores consome de um tópico.	Estimativa de tempo (em segundos) para drenar o atraso no deslocamento da partição.
`OffsetLag`	Depois que o grupo de consumidores consome de um tópico.	Atraso do consumidor no nível de partição em número de deslocamentos.

Entenda os estados do cluster provisionado pelo MSK

A tabela a seguir mostra os possíveis estados de um cluster provisionado pelo MSK e descreve o que eles significam. A menos que especificado de outra forma, os estados de cluster provisionado do MSK se aplicam aos tipos de broker Standard e Express. Essa tabela também descreve quais ações você pode e não pode realizar quando um cluster provisionado pelo MSK está em um desses estados. Para descobrir o estado de um cluster, você pode acessar o AWS Management Console.

Você também pode usar o comando describe-cluster-v2 ou a operação DescribeClusterV2 para descrever o cluster provisionado. A descrição de um cluster inclui seu estado.

Estado do cluster provisionado MSK	Significado e ações possíveis
ACTIVE	Você pode produzir e consumir dados. Você também pode realizar AWS CLI operações e APIs do Amazon MSK no cluster.
CRIANDO	O Amazon MSK está configurando o cluster provisionado. Você deve esperar que o cluster alcance o estado ATIVO antes de poder usá-lo para produzir ou consumir dados ou para executar a API do Amazon MSK ou AWS CLI operações neles.
EXCLUINDO	O cluster provisionado está sendo excluído. Você não pode usá-lo para produzir ou consumir dados. Você também não pode executar a API do Amazon MSK ou AWS CLI operações nela.
COM FALHA	O processo de criação ou exclusão do cluster provisionado falhou. Você não pode usar o cluster para produzir ou consumir dados. Você pode excluir o cluster, mas não pode executar a API Amazon MSK nem AWS CLI atualizar operações nele.
HEALING	O Amazon MSK está executando uma operação interna, como a substituição de um agente não íntegro. Por exemplo, talvez o agente não esteja respondendo. Você ainda pode usar o cluster provisionado para produzir e consumir dados. No entanto, você não pode realizar operações de API ou AWS CLI atualizar a API do Amazon MSK no cluster até que ele retorne ao estado ATIVO.

Estado do cluster provisionado MSK	Significado e ações possíveis
MAINTENANCE	(Somente corretores padrão) O Amazon MSK está realizando operações de manutenção de rotina no cluster. Essas operações de manutenção incluem a aplicação de patches de segurança. Você ainda pode usar o cluster para produzir e consumir dados. No entanto, você não pode realizar operações de atualização da API ou AWS CLI do Amazon MSK no cluster até que ele retorne ao estado ATIVO. O estado do cluster permanece ATIVO durante a manutenção nos corretores Express. Consulte Aplicação de patches.
REBOOTING_BROKER	O Amazon MSK está reiniciando um agente. Você ainda pode usar o cluster provisionado para produzir e consumir dados. No entanto, você não pode realizar operações de API ou AWS CLI atualizar a API do Amazon MSK no cluster até que ele retorne ao estado ATIVO.
ATUALIZANDO	Uma API ou AWS CLI operação do Amazon MSK iniciada pelo usuário está atualizando o cluster provisionado. Você ainda pode usar o cluster provisionado para produzir e consumir dados. No entanto, você não pode realizar nenhuma operação adicional de API ou AWS CLI atualização do Amazon MSK no cluster até que ele retorne ao estado ATIVO.

Métricas do Amazon MSK para monitorar corretores Express com CloudWatch

O Amazon MSK se integra CloudWatch para que você possa coletar, visualizar e analisar CloudWatch métricas para seus corretores MSK Express. As métricas que você configura para seus clusters provisionados pelo MSK são coletadas e enviadas automaticamente em intervalos de 1 CloudWatch minuto. Você pode definir o nível de monitoramento de um cluster provisionado

pelo MSK como um dos seguintes:`DEFAULT`,,`PER_BROKER`, `PER_TOPIC_PER_BROKER` ou. `PER_TOPIC_PER_PARTITION` As tabelas nas seções a seguir mostram as métricas que estão disponíveis a partir de cada nível de monitoramento.

As métricas no nível `DEFAULT` são gratuitas. Os preços de outras métricas estão descritos na página de CloudWatchpreços da Amazon.

`DEFAULT`Monitoramento de nível para corretores Express

As métricas descritas na tabela a seguir estão disponíveis gratuitamente no nível de `DEFAULT` monitoramento.

Name	Quando visível	Dimensões	Descrição
ActiveControllerCount	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster	Somente um controlador por cluster deve estar ativo em qualquer momento.
BytesInPerSec	Depois de criar um tópico.	Nome do cluster, ID do agente, tópico	O número de bytes por segundo recebidos dos clientes. Essa métrica está disponível por agente e também por tópico.
BytesOutPerSec	Depois de criar um tópico.	Nome do cluster, ID do agente, tópico	O número de bytes por segundo enviados aos clientes. Essa métrica está disponível por agente e também por tópico.
ClientConnectionCount	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente, autenticação de cliente	O número de conexões de cliente autenticadas e ativas.

Name	Quando visível	Dimensões	Descrição
ConnectionCount	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de conexões ativas autenticadas, não autenticadas e entre agentes.
CpuIdle	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	A porcentagem de tempo ocioso da CPU.
CpuSystem	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	A porcentagem de CPU no espaço do kernel.
CpuUser	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	A porcentagem de CPU no espaço do usuário.
GlobalPartitionCount	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster	O número de partições em todos os tópicos no cluster, excluindo réplicas. Como `GlobalPartitionCount` não inclui réplicas, a soma dos `PartitionCount` valores pode ser maior do que `GlobalPartitionCount` se o fator de replicação de um tópico for maior que. 1

Name	Quando visível	Dimensões	Descrição
GlobalTopicCount	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster	Número total de tópicos em todos os agentes no cluster.
EstimatedMaxTimeLag	Depois que o grupo de consumidores consome de um tópico.	Grupo de consumidores, tópico	Estimativa de tempo (em segundos) para drenar `MaxOffsetLag` .
LeaderCount	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número total de líderes de partições por agente, sem incluir réplicas.
MaxOffsetLag	Depois que o grupo de consumidores consome de um tópico.	Grupo de consumidores, tópico	O atraso máximo de deslocamento entre todas as partições em um tópico.
MemoryBuffered	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O tamanho, em bytes, da memória armazenada em buffer para o agente.
MemoryCached	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O tamanho, em bytes, da memória armazenada em cache para o agente.
MemoryFree	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O tamanho, em bytes, de memória que é gratuita e disponível para o agente.

Name	Quando visível	Dimensões	Descrição
MemoryUsed	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O tamanho, em bytes, de memória que está em uso pelo agente.
MessagesInPerSec	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de mensagens recebidas por segundo do agente.
NetworkRxDropped	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de pacotes de recebimento descartados.
NetworkRxErrors	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de erros de recepção da rede para o agente.
NetworkRxPackets	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de pacotes recebidos pelo agente.
NetworkTxDropped	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de pacotes de transmissão descartados.
NetworkTxErrors	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de erros de transmissão da rede para o agente.
NetworkTxPackets	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número de pacotes transmitidos pelo agente.
PartitionCount	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número total de partições de tópico por agente, incluindo réplicas.

Name	Quando visível	Dimensões	Descrição
ProduceTotalTimeMs Mean	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O tempo médio de produção em milissegundos.
RequestBytesMean	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	O número médio de bytes de solicitações do agente.
RequestTime	Após a limitação da solicitação ser aplicada.	Nome do cluster, ID do agente	O tempo médio em milissegundos gasto na rede do agente e nos I/O threads para processar solicitaç ões.
StorageUsed	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster	O armazenamento total usado em todas as partições do cluster, excluindo as réplicas.
SumOffsetLag	Depois que o grupo de consumidores consome de um tópico.	Grupo de consumido res, tópico	O atraso de deslocamento agregado para todas as partições em um tópico.
UserPartitionExists	Depois que o cluster passa para o estado ACTIVE.	Nome do cluster, ID do agente	Métrica booleana que indica a presença de uma partição de propriedade do usuário em uma corretora. Um valor de 1 indica a presença de partições no corretor.

PER_BROKER Monitoramento de nível para corretores Express

Ao definir o nível de monitoramento como PER_BROKER, você obtém as métricas descritas na tabela a seguir, além de todas as métricas de nível DEFAULT. Você paga pelas métricas na tabela a seguir, enquanto as métricas de DEFAULT nível continuam gratuitas. As métricas nesta tabela têm as seguintes dimensões: nome do cluster, ID do agente.

Métricas adicionais disponíveis a partir do nível de monitoramento PER_BROKER

Name	Quando visível	Descrição
ConnectionCloseRate	Depois que o cluster passa para o estado ACTIVE.	O número de conexões fechadas por segundo por receptor. Esse número é agregado por receptor e filtrado para os receptores do cliente.
ConnectionCreationRate	Depois que o cluster passa para o estado ACTIVE.	O número de novas conexões estabelecidas por segundo por receptor. Esse número é agregado por receptor e filtrado para os receptores do cliente.
FetchConsumerLocal TimeMsMean	Depois de haver um produtor/ consumidor.	O tempo médio em milissegu ndos que a solicitação do consumidor é processada no líder.
FetchConsumerReque stQueueTimeMsMean	Depois de haver um produtor/ consumidor.	O tempo médio em milissegu ndos que a solicitação do consumidor aguarda na fila de solicitações.
FetchConsumerRespo nseQueueTimeMsMean	Depois de haver um produtor/ consumidor.	O tempo médio em milissegu ndos que a solicitação do consumidor aguarda na fila de resposta.

Name	Quando visível	Descrição
FetchConsumerResponseSendTimeMsMean	Depois de haver um produtor/consumidor.	O tempo médio, em milissegundos, para que o consumidor envie uma resposta.
FetchConsumerTotalTimeMsMean	Depois de haver um produtor/consumidor.	O tempo total médio em milissegundos que os consumidores gastam obtendo dados do agente.
FetchFollowerLocalTimeMsMean	Depois de haver um produtor/consumidor.	O tempo médio em milissegundos que a solicitação do seguidor é processada no líder.
FetchFollowerRequestQueueTimeMsMean	Depois de haver um produtor/consumidor.	O tempo médio em milissegundos que a solicitação de seguidor aguarda na fila de solicitações.
FetchFollowerResponseQueueTimeMsMean	Depois de haver um produtor/consumidor.	O tempo médio em milissegundos que a solicitação de seguidor aguarda na fila de resposta.
FetchFollowerResponseSendTimeMsMean	Depois de haver um produtor/consumidor.	O tempo médio em milissegundos para o seguidor enviar uma resposta.
FetchFollowerTotalTimeMsMean	Depois de haver um produtor/consumidor.	O tempo total médio em milissegundos que os seguidores gastam obtendo e dados do agente.
FetchThrottleByteRate	Depois que a limitação da largura de banda é aplicada.	O número de bytes limitados por segundo.

Name	Quando visível	Descrição
FetchThrottleQueueSize	Depois que a limitação da largura de banda é aplicada.	O número de mensagens na fila de limitação.
FetchThrottleTime	Depois que a limitação da largura de banda é aplicada.	O tempo médio de limitações de busca em milissegundos.
IAMNumberOfConnectionRequests	Depois que o cluster passa para o estado ACTIVE.	O número de solicitações de autenticação do IAM por segundo.
IAMTooManyConnections	Depois que o cluster passa para o estado ACTIVE.	O número de conexões tentadas além de 100. `0`significa que o número de conexões está dentro do limite. Se `>0` o limite do acelerador estiver sendo excedido e você precisar reduzir o número de conexões.
NetworkProcessorAvgIdlePercent	Depois que o cluster passa para o estado ACTIVE.	A porcentagem média do tempo em que os processadores de rede estão ociosos.
ProduceLocalTimeMsMean	Depois que o cluster passa para o estado ACTIVE.	O tempo médio em milissegundos que a solicitação leva para ser processada no líder.
ProduceRequestQueueTimeMsMean	Depois que o cluster passa para o estado ACTIVE.	O tempo médio em milissegundos que as mensagens de solicitação gastam na fila.
ProduceResponseQueueTimeMsMean	Depois que o cluster passa para o estado ACTIVE.	O tempo médio em milissegundos que as mensagens de resposta gastam na fila.

Name	Quando visível	Descrição
ProduceResponseSendTimeMsMean	Depois que o cluster passa para o estado ACTIVE.	O tempo médio em milissegundos gasto no envio de mensagens de resposta.
ProduceThrottleByteRate	Depois que a limitação da largura de banda é aplicada.	O número de bytes limitados por segundo.
ProduceThrottleQueueSize	Depois que a limitação da largura de banda é aplicada.	O número de mensagens na fila de limitação.
ProduceThrottleTime	Depois que a limitação da largura de banda é aplicada.	O tempo médio de limitação da produção em milissegundos.
ProduceTotalTimeMsMean	Depois que o cluster passa para o estado ACTIVE.	O tempo médio de produção em milissegundos.
ReplicationBytesInPerSec	Depois de criar um tópico.	O número de bytes por segundo recebidos dos outros agentes.
ReplicationBytesOutPerSec	Depois de criar um tópico.	O número de bytes por segundo enviados para outros agentes.
RequestExemptFromThrottleTime	Após a limitação da solicitação ser aplicada.	O tempo médio em milissegundos gasto na rede do broker e nos I/O threads para processar solicitações isentas de limitação.
RequestHandlerAvgIdlePercent	Depois que o cluster passa para o estado ACTIVE.	A porcentagem média do tempo em que os threads do manipulador de solicitações estão ociosos.

Name	Quando visível	Descrição
RequestThrottleQueueSize	Após a limitação da solicitação ser aplicada.	O número de mensagens na fila de limitação.
RequestThrottleTime	Após a limitação da solicitação ser aplicada.	O tempo médio da limitação de solicitações em milissegundos.
TcpConnections	Depois que o cluster passa para o estado ACTIVE.	Mostra o número de segmentos TCP de entrada e saída com o sinalizador SYN definido.
TrafficBytes	Depois que o cluster passa para o estado ACTIVE.	Mostra o tráfego de rede em bytes gerais entre clientes (produtores e consumidores) e agentes. O tráfego entre agentes não é relatado.

`PER_TOPIC_PER_PARTITION`monitoramento de nível para corretores Express

Ao definir o nível de monitoramento como`PER_TOPIC_PER_PARTITION`, você obtém as métricas descritas na tabela a seguir, além de todas as métricas dos `DEFAULT` níveis `PER_TOPIC_PER_BROKER`PER_BROKER, e. Somente as métricas de `DEFAULT` nível são gratuitas. As métricas nesta tabela têm as seguintes dimensões: grupo de consumidores, tópico, partição.

Métricas adicionais disponíveis a partir do nível de monitoramento PER_PARTITION

Name	Quando visível	Descrição
EstimatedTimeLag	Depois que o grupo de consumidores consome de um tópico.	Estimativa de tempo (em segundos) para drenar o atraso no deslocamento da partição.

Name	Quando visível	Descrição
OffsetLag	Depois que o grupo de consumidores consome de um tópico.	Atraso do consumidor no nível de partição em número de deslocamentos.

`PER_TOPIC_PER_BROKER`monitoramento de nível para corretores Express

Ao definir o nível de monitoramento como`PER_TOPIC_PER_BROKER`, você obtém as métricas descritas na tabela a seguir, além de todas as métricas dos `DEFAULT` níveis `PER_BROKER` e. Somente as métricas de `DEFAULT` nível são gratuitas. As métricas nesta tabela têm as seguintes dimensões: nome do cluster, ID do agente, tópico.

> **Important**
>
> As métricas na tabela a seguir aparecem somente depois que seus valores se tornam diferentes de zero pela primeira vez. Por exemplo, para ver BytesInPerSec, um ou mais produtores devem primeiro enviar dados para o cluster.

Métricas adicionais disponíveis a partir do nível de monitoramento PER_TOPIC_PER_BROKER

Name	Quando visível	Descrição
MessagesInPerSec	Depois de criar um tópico.	O número de mensagens recebidas por segundo.

Monitore um cluster provisionado pelo MSK com o Prometheus

Você pode monitorar seu cluster provisionado pelo MSK com o Prometheus, um sistema de monitoramento de código aberto para dados métricos de séries temporais. Você pode publicar esses dados no Amazon Managed Service for Prometheus usando o recurso de gravação remota do Prometheus. Você também pode usar ferramentas compatíveis com métricas formatadas pelo Prometheus ou ferramentas que se integram ao Amazon MSK Open Monitoring, como Datadog, Lenses, New Relic e Sumo logic. O monitoramento aberto está disponível gratuitamente, mas cobranças são aplicáveis à transferência de dados entre zonas de disponibilidade.

Para obter informações sobre o Prometheus, consulte a documentação do Prometheus.

Para obter informações sobre o uso do Prometheus, consulte Aprimorar insights operacionais para o Amazon MSK usando o Amazon Managed Service para Prometheus e o Amazon Managed Grafana.

> **Note**
>
> KRaft o modo de metadados e os corretores MSK Express não podem ter o monitoramento aberto e o acesso público habilitados.

Habilite o monitoramento aberto em novos clusters provisionados pelo MSK

Este procedimento descreve como habilitar o monitoramento aberto em um novo cluster MSK usando a AWS Management Console AWS CLI, a ou a API Amazon MSK.

Usando o AWS Management Console

1. Faça login no AWS Management Console e abra o console do Amazon MSK em https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. Na seção Monitoring (Monitoramento), marque a caixa de seleção ao lado de Enable open monitoring with Prometheus (Habilitar o monitoramento aberto com o Prometheus).
3. Forneça as informações obrigatórias em todas as seções da página e revise todas as opções disponíveis.
4. Selecione Criar cluster.

Usando o AWS CLI

- Invoque o comando create-cluster e especifique a opção `open-monitoring`. Habilite o `JmxExporter`, o `NodeExporter` ou ambos. Se você especificar o `open-monitoring`, os dois exportadores não poderão ser desabilitados ao mesmo tempo.

Uso da API

- Invoque a CreateClusteroperação e especifique`OpenMonitoring`. Habilite o `jmxExporter`, o `nodeExporter` ou ambos. Se você especificar o `OpenMonitoring`, os dois exportadores não poderão ser desabilitados ao mesmo tempo.

Habilite o monitoramento aberto no cluster provisionado MSK existente

Para ativar o monitoramento aberto, certifique-se de que o cluster provisionado do MSK esteja no estado. `ACTIVE`

Usando o AWS Management Console

1. Faça login no AWS Management Console e abra o console do Amazon MSK em https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. Escolha o nome do cluster que deseja atualizar. Você será redirecionado para uma página com os detalhes do cluster.
3. Na guia Propriedades, role para baixo para encontrar a seção Monitoramento.
4. Escolha Editar.
5. Marque a caixa de seleção ao lado de Enable open monitoring with Prometheus (Habilitar o monitoramento aberto com o Prometheus).
6. Escolha Salvar alterações.

Usando o AWS CLI

- Invoque o comando update-monitoring e especifique a opção `open-monitoring`. Habilite o `JmxExporter`, o `NodeExporter` ou ambos. Se você especificar o `open-monitoring`, os dois exportadores não poderão ser desabilitados ao mesmo tempo.

Uso da API

- Invoque a UpdateMonitoringoperação e especifique`OpenMonitoring`. Habilite o `jmxExporter`, o `nodeExporter` ou ambos. Se você especificar o `OpenMonitoring`, os dois exportadores não poderão ser desabilitados ao mesmo tempo.

Configurar um host Prometheus em uma instância da Amazon EC2

Este procedimento descreve como configurar um host Prometheus usando um arquivo prometheus.yml.

1. Faça o download do servidor Prometheus https://prometheus.io/download/#prometheus para sua instância da Amazon. EC2
2. Extraia o arquivo obtido por download para um diretório e acesse esse diretório.

3. Crie um arquivo com o seguinte conteúdo e nomeie-o como `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. Use a ListNodesoperação para obter uma lista dos corretores do seu cluster.

5. Crie um arquivo denominado `targets.json` com a seguinte JSON: Substitua *broker_dns_1broker_dns_2*,, e o resto dos nomes DNS dos corretores pelos nomes DNS que você obteve para seus corretores na etapa anterior. Inclua todos os agentes que você obteve na etapa anterior. O Amazon MSK usa a porta 11001 para o JMX Exporter e a porta 11002 para o Node Exporter.

   ZooKeeper mode targets.json

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

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

KRaft mode targets.json

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

```
      .
      .
      .
      "broker_dns_N:11002"
    ]
  }
]
```

> **Note**
>
> Para extrair métricas do JMX dos KRaft controladores, adicione nomes DNS do controlador como destinos no arquivo JSON. Por exemplo: `controller_dns_1:11001`, substituindo `controller_dns_1` pelo nome DNS real do controlador.

6. Para iniciar o servidor Prometheus em sua instância EC2 Amazon, execute o seguinte comando no diretório em que você extraiu os arquivos do Prometheus e salvou e. `prometheus.yml` `targets.json`

```
./prometheus
```

7. Encontre o endereço IP IPv4 público da EC2 instância da Amazon em que você executou o Prometheus na etapa anterior. Esse endereço IP público será necessário na próxima etapa.
8. Para acessar a interface web do Prometheus, abra um navegador que possa acessar sua instância da EC2 Amazon e acesse*Prometheus-Instance-Public-IP*`:9090`, *Prometheus-Instance-Public-IP* onde está o endereço IP público que você obteve na etapa anterior.

Usar métricas do Prometheus

Todas as métricas emitidas pelo Apache Kafka para o JMX são acessíveis ao usar o monitoramento aberto com o Prometheus. Para obter informações sobre as métricas do Apache Kafka, consulte Monitoring (Monitoramento) na documentação do Apache Kafka. Junto com as métricas do Apache Kafka, as métricas de atraso do consumidor também estão disponíveis na porta 11001 com o nome JMX. MBean `kafka.consumer.group:type=ConsumerLagMetrics` Você também pode usar o Prometheus Node Exporter para obter métricas de CPU e disco para seus agentes na porta 11002.

Armazenar as métricas do Prometheus no Amazon Managed Service for Prometheus

O Amazon Managed Service for Prometheus é um serviço de monitoramento e emissão de alertas compatível com o Prometheus que você pode usar para monitorar os clusters do Amazon MSK. É um serviço totalmente gerenciado que dimensiona automaticamente a ingestão, o armazenamento, a consulta e o alerta de métricas. Ele também se integra aos serviços de AWS segurança para oferecer acesso rápido e seguro aos seus dados. É possível usar a linguagem de consulta PromQL de código aberto para consultar suas métricas e emitir alertas sobre elas.

Para obter mais informações, consulte Conceitos básicos do Amazon Managed Service for Prometheus.

Monitorar atrasos do consumidor

O monitoramento do atraso do consumidor permite identificar consumidores lentos ou presos que não estão acompanhando os dados mais recentes disponíveis em um tópico. Quando necessário, você poderá adotar medidas corretivas, como escalar ou reinicializar esses consumidores. Para monitorar o atraso do consumidor, você pode usar a Amazon CloudWatch ou abrir o monitoramento com o Prometheus.

As métricas de atraso do consumidor quantificam a diferença entre os dados mais recentes gravados em seus tópicos e os dados lidos por suas aplicações. O Amazon MSK fornece as seguintes métricas de atraso do consumidor, que você pode obter por meio da Amazon CloudWatch ou por meio do monitoramento aberto com o Prometheus:,,, e. `EstimatedMaxTimeLag` `EstimatedTimeLag` `MaxOffsetLag` `OffsetLag` `SumOffsetLag` Para obter informações sobre essas métricas, consulte the section called “CloudWatch métricas para corretores padrão”.

O Amazon MSK é compatível com métricas de atraso do consumidor para clusters com o Apache Kafka 2.2.1 ou versões posteriores. Considere os seguintes pontos ao trabalhar com o Kafka e CloudWatch as métricas:

- As métricas de atraso do consumidor são emitidas somente se um grupo de consumidores estiver em um estado ESTÁVEL ou VAZIO. Um grupo de consumidores fica ESTÁVEL após a conclusão com êxito do rebalanceamento, garantindo que as partições sejam distribuídas uniformemente entre os consumidores.
- As métricas de atraso do consumidor estão ausentes nos seguintes cenários:
  - Se o grupo de consumidores estiver instável.
  - O nome do grupo de consumidores contém dois pontos (:).

  - Você não definiu a compensação do consumidor para o grupo de consumidores.
- Os nomes dos grupos de consumidores são usados como dimensões para as métricas de atraso do consumidor em CloudWatch. Enquanto o Kafka suporta caracteres UTF-8 em nomes de grupos de consumidores, CloudWatch suporta somente caracteres ASCII para valores de dimensão. Se você usar caracteres não ASCII em nomes de grupos de consumidores, CloudWatch descarta as métricas de atraso do consumidor. Para garantir que suas métricas de atraso do consumidor sejam capturadas corretamente CloudWatch, você deve usar somente caracteres ASCII nos nomes dos grupos de consumidores.

Usar alertas de capacidade de armazenamento do Amazon MSK

Nos clusters provisionados pelo Amazon MSK, você escolhe a capacidade de armazenamento principal do cluster. O esgotamento da capacidade de armazenamento de um agente no cluster provisionado pode afetar a capacidade do cluster de produzir e consumir dados, resultando em um tempo de inatividade dispendioso. O Amazon MSK oferece CloudWatch métricas para ajudar você a monitorar a capacidade de armazenamento do seu cluster. No entanto, para facilitar a detecção e a resolução de problemas de capacidade de armazenamento, o Amazon MSK envia automaticamente alertas dinâmicos de capacidade de armazenamento do cluster. Os alertas de capacidade de armazenamento incluem recomendações para etapas de curto e longo prazo para o gerenciamento da capacidade de armazenamento do cluster. No console do Amazon MSK, você pode usar links rápidos nos alertas para executar imediatamente as ações recomendadas.

Há dois tipos de alertas de capacidade de armazenamento do MSK: proativos e corretivos.

- Alertas proativos (“Ação necessária”) de capacidade de armazenamento avisam você sobre possíveis problemas de armazenamento no cluster. Quando um agente em um cluster do MSK usar mais de 60% ou 80% da capacidade de armazenamento em disco, você receberá alertas proativos para o agente afetado.
- Os alertas de capacidade de armazenamento corretivos (“Ação crítica necessária”) exigem que você tome medidas corretivas para corrigir um problema crítico no cluster quando um dos agentes do cluster do MSK fica sem capacidade de armazenamento em disco.

O Amazon MSK envia automaticamente esses alertas para o console do Amazon MSK, AWS Health Dashboard EventBridge, Amazon e contatos de e-mail da sua AWS conta. Você também pode configurar EventBridge a Amazon para entregar esses alertas ao Slack ou a ferramentas como New Relic e Datadog.

Os alertas de capacidade de armazenamento são habilitados por padrão para todos os clusters provisionados pelo MSK e não podem ser desativados. Esse recurso é compatível em todas as regiões em que o MSK está disponível.

Monitorar alertas de capacidade de armazenamento

Você pode verificar os alertas de capacidade de armazenamento de várias maneiras:

- Vá para o console do Amazon MSK. Os alertas de capacidade de armazenamento são exibidos no painel de alertas do cluster por 90 dias. Os alertas contêm recomendações e ações de link com um único clique para resolver problemas de capacidade de armazenamento em disco.
- Use ListClustersListClustersV2 ou DescribeClusterV2 APIs para visualizar `CustomerActionStatus` todos os alertas de um cluster. DescribeCluster
- Acesse o AWS Health Dashboard para ver os alertas do MSK e de outros serviços da AWS .
- Configure a AWS Health API e EventBridge a Amazon para encaminhar notificações de alerta para plataformas de terceiros NewRelic, como Datadog e Slack.

Atualizar as configurações de segurança de um cluster do Amazon MSK

Use a operação UpdateSecurityAmazon MSK para atualizar as configurações de autenticação e criptografia cliente-agente do seu cluster MSK. Você também pode atualizar a Autoridade de Segurança Privada usada para assinar certificados para autenticação TLS mútua. Você não pode alterar a configuração de criptografia no cluster (broker-to-broker).

O cluster deve estar no estado `ACTIVE` para que você atualize suas configurações de segurança.

Se você ativar a autenticação usando IAM, SASL ou TLS, também deverá ativar a criptografia entre clientes e agentes. A tabela a seguir mostra as combinações possíveis.

Autenticação	Opções de criptografia cliente-agente	Criptografia agente-agente
Unauthenticated	TLS, PLAINTEXT, TLS_PLAINTEXT	Pode estar ativado ou desativado.
mTLS	TLS, TLS_PLAINTEXT	Precisa estar ativado.
SASL/SCRAM	TLS	Precisa estar ativado.

Autenticação	Opções de criptografia cliente-agente	Criptografia agente-agente
SASL/IAM	TLS	Precisa estar ativado.

Quando a criptografia cliente-agente estiver definida como `TLS_PLAINTEXT` e a autenticação do cliente estiver definida como `mTLS`, o Amazon MSK criará dois tipos de receptores aos quais os clientes se conectarão: um ouvinte para os clientes se conectarem usando a autenticação mTLS com criptografia TLS e outro para os clientes se conectarem sem autenticação ou criptografia (texto simples).

Para obter mais informações sobre as configurações de segurança, consulte the section called "Segurança".

Atualize as configurações de segurança do cluster Amazon MSK usando o AWS Management Console

1. Faça login no AWS Management Console e abra o console do Amazon MSK em https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. Selecione o cluster do MSK que deseja atualizar.
3. Na seção Configurações de segurança, escolha Editar.
4. Escolha as configurações de autenticação e criptografia que deseja aplicar para o cluster e, em seguida, escolha Salvar alterações.

Atualizando as configurações de segurança do cluster Amazon MSK usando o AWS CLI

1. Crie um arquivo JSON contendo as configurações de criptografia desejadas para o cluster. Veja um exemplo a seguir.

> **Note**
>
> Você só pode atualizar a configuração de criptografia cliente-agente. Você não pode atualizar a configuração de criptografia no cluster (broker-to-broker).

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

2. Crie um arquivo JSON contendo as configurações de autenticação desejadas para o cluster. Veja um exemplo a seguir.

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

3. Execute o seguinte 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
```

A saída dessa operação `update-security` é semelhante ao seguinte 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 o status da `update-security` operação, execute o comando a seguir, *`ClusterOperationArn`* substituindo-o pelo ARN obtido na saída do `update-security` comando.

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

A saída desse comando `describe-cluster-operation` é semelhante ao seguinte JSON de exemplo.

```
{
    "ClusterOperationInfo": {
        "ClientRequestId": "c0b7af47-8591-45b5-9c0c-909a1a2c99ea",
        "ClusterArn": "arn:aws:kafka:us-east-1:012345678012:cluster/
exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2",
        "CreationTime": "2021-09-17T02:35:47.753000+00:00",
```

```
        "OperationArn": "arn:aws:kafka:us-east-1:012345678012:cluster-
operation/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2/0123abcd-
abcd-4f7f-1234-9876543210ef",
        "OperationState": "PENDING",
        "OperationType": "UPDATE_SECURITY",
        "SourceClusterInfo": {},
        "TargetClusterInfo": {}
    }
}
```

Se `OperationState` tiver o valor `PENDING` ou `UPDATE_IN_PROGRESS`, aguarde um pouco e execute o comando `describe-cluster-operation` novamente.

> **Note**
>
> As operações de API AWS CLI e de atualização das configurações de segurança de um cluster são idempotentes. Isso significa que se você invocar a operação de atualização de segurança e especificar uma configuração de autenticação ou criptografia que seja a mesma configuração que o cluster já tem, essa configuração não será alterada.

Atualizar as configurações de segurança de um cluster usando a API

Para atualizar as configurações de segurança de um cluster Amazon MSK usando a API, consulte UpdateSecurity.

> **Note**
>
> As operações de API AWS CLI e de atualização das configurações de segurança de um cluster MSK são idempotentes. Isso significa que se você invocar a operação de atualização de segurança e especificar uma configuração de autenticação ou criptografia que seja a mesma configuração que o cluster já tem, essa configuração não será alterada.

Expandir o número de agentes em um cluster do Amazon MSK

Execute esta operação do Amazon MSK quando você quiser aumentar o número de agentes em seu cluster do MSK. Para expandir um cluster, certifique-se de que ele esteja no estado `ACTIVE`.

> **Important**
>
> Certifique-se de usar esta operação do Amazon MSK se quiser expandir um cluster do MSK. Não tente adicionar agentes a um cluster sem usar essa operação.

Para obter informações sobre como reequilibrar partições depois de adicionar agentes a um cluster, consulte the section called “Reatribuir partições”.

Expanda um cluster Amazon MSK usando o AWS Management Console

Este processo descreve como aumentar o número de agentes em um cluster do Amazon MSK usando o AWS Management Console.

1. Faça login no AWS Management Console e abra o console do Amazon MSK em https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. Escolha o cluster do MSK cujo número de agentes deseja aumentar.
3. No menu suspenso Ações, escolha Editar número de corretores.
4. Insira o número de agentes que você deseja que o cluster tenha por zona de disponibilidade e escolha Salvar alterações.

Expanda um cluster Amazon MSK usando o AWS CLI

Este processo descreve como aumentar o número de agentes em um cluster do Amazon MSK usando o AWS CLI.

1. Execute o comando a seguir, substituindo *`ClusterArn`* pelo nome do recurso da Amazon (ARN) que você obteve quando criou o cluster. Se você não tiver o ARN do cluster, poderá encontrá-lo listando todos os clusters. Para obter mais informações, consulte the section called “Listar clusters”.

   Substitua *`Current-Cluster-Version`* pela versão atual do cluster.

> **Important**
>
> As versões de cluster não são inteiros simples. Para encontrar a versão atual do cluster, use a DescribeClusteroperação ou o comando AWS CLI describe-cluster. Uma versão de exemplo é KTVPDKIKX0DER.

O *Target-Number-of-Brokers* parâmetro representa o número total de nós de intermediários que você deseja que o cluster tenha quando essa operação for concluída com êxito. O valor especificado *Target-Number-of-Brokers* deve ser um número inteiro maior que o número atual de corretores no cluster. Também deve ser um múltiplo do número de zonas de disponibilidade.

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

A saída dessa operação `update-broker-count` é semelhante ao seguinte 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 obter o resultado da `update-broker-count` operação, execute o comando a seguir, *ClusterOperationArn* substituindo-o pelo ARN obtido na saída do `update-broker-count` comando.

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

A saída desse comando `describe-cluster-operation` é semelhante ao seguinte JSON de exemplo.

```
{
    "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
        }
    }
}
```

Nesta saída, `OperationType` é `INCREASE_BROKER_COUNT`. Se `OperationState` tiver o valor `UPDATE_IN_PROGRESS`, aguarde um pouco e execute o comando `describe-cluster-operation` novamente.

Expandir um cluster do Amazon MSK usando a API

Para aumentar o número de corretores em um cluster usando a API, consulte UpdateBrokerCount.

Remover um agente de um cluster do Amazon MSK

Use esta operação do Amazon MSK quando quiser remover agentes dos clusters provisionados do Amazon Managed Streaming for Apache Kafka (MSK). Você pode reduzir a capacidade de armazenamento e computação do cluster removendo conjuntos de agentes, sem impacto na disponibilidade, risco de durabilidade de dados ou interrupção nas aplicações de fluxo de dados.

Você pode adicionar mais agentes ao cluster para lidar com o aumento do tráfego e remover agentes quando o tráfego diminuir. Com a capacidade de adição e remoção de agentes, você pode utilizar melhor a capacidade do cluster e otimizar os custos de infraestrutura do MSK. A remoção do agente lhe dá o controle no nível do agente sobre a capacidade existente do cluster para atender às suas necessidades de workload e evitar a migração para outro cluster.

Use o AWS console, a interface de linha de comando (CLI), o SDK ou AWS CloudFormation para reduzir o número de agentes do seu cluster provisionado. O MSK escolhe os agentes que não têm

nenhuma partição neles (exceto os tópicos de canário) e impede que as aplicações produzam dados para esses agentes, ao mesmo tempo que os remove com segurança do cluster.

Você deve remover um agente por zona de disponibilidade, caso queira reduzir o armazenamento e a computação de um cluster. Por exemplo, você pode remover dois agentes de um cluster de duas zonas de disponibilidade, ou três agentes de um cluster de três zonas de disponibilidade em uma única operação de remoção de agentes.

Para obter informações sobre como rebalancear as partições depois de remover agentes de um cluster, consulte the section called “Reatribuir partições”.

Você pode remover os agentes de todos os clusters provisionados do MSK baseados em M5 e M7g, independentemente do tamanho da instância.

A remoção do broker é suportada nas versões 2.8.1 e superiores do Kafka, inclusive nos KRaft clusters de modo.

Tópicos

Prepare-se para remover os agentes ao remover todas as partições

Antes de iniciar o processo de remoção do agente, primeiro mova todas as partições, exceto aquelas dos tópicos `__amazon_msk_canary` e `__amazon_msk_canary_state` dos agentes que você planeja remover. Trata-se de tópicos internos que o Amazon MSK cria para métricas de integridade e diagnóstico do cluster.

Você pode usar o Kafka admin APIs ou o Cruise Control para mover partições para outros corretores que você pretende manter no cluster. Consulte Reatribuir partições.

Exemplo de processo para remover partições

Esta seção é um exemplo de como remover partições do agente que você pretende remover. Suponha que você tenha um cluster com seis agentes, dois agentes em cada AZ, e ele tenha quatro tópicos:

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

1. Crie uma máquina cliente conforme descrito em Criar uma máquina cliente.
2. Depois de configurar a máquina cliente, execute o comando a seguir para listar todos os tópicos disponíveis no cluster.

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

   Neste exemplo, vemos quatro nomes de tópicos: `__amazon_msk_canary`, `__consumer_offsets`, `__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-c657f7e4ff32-2` e `msk-brk-rmv`.
3. Crie um arquivo json chamado `topics.json` na máquina cliente e adicione todos os nomes dos tópicos do usuário, como no exemplo de código a seguir. Você não precisa incluir o nome do tópico `__amazon_msk_canary`, pois é um tópico gerenciado pelo serviço que será movido automaticamente quando necessário.

   ```
   {
   "topics": [
   {"topic": "msk-brk-rmv"},
   {"topic": "__consumer_offsets"},
   {"topic": "__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-
   c657f7e4ff32-2"}
   ],
   "version":1
   }
   ```
4. Execute o comando a seguir para gerar uma proposta para mover partições para apenas três agentes dos seis agentes no cluster.

   ```
   ./bin/kafka-reassign-partitions.sh --bootstrap-server “CLUSTER_BOOTSTRAP_STRING” --
   topics-to-move-json-file topics.json --broker-list 1,2,3 --generate
   ```
5. Crie um arquivo chamado `reassignment-file.json` e copie a `proposed partition reassignment configuration` que você obteve do comando acima.

6. Execute o comando a seguir para mover as partições que você especificou em `reassignment-file.json`.

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

   A saída será semelhante à seguinte:

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

7. Execute o comando a seguir para verificar se todas as partições foram movidas.

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

   A saída será semelhante à seguinte. Monitore o status até que todas as partições nos tópicos solicitados tenham sido reatribuídas com êxito:

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

8. Quando o status indicar que a reatribuição de partição de cada partição foi concluída, monitore as métricas `UserPartitionExists` por cinco minutos para garantir que elas exibam `0` para os agentes dos quais você moveu as partições. Depois de confirmar essa questão, você pode prosseguir para remover o agente do cluster.

Remover um corretor com o AWS Management Console

Para remover corretores com o AWS Management Console

1. Abra o console do Amazon MSK em https://console.aws.amazon.com/msk/.
2. Escolha o cluster do MSK que contém os agentes que deseja remover.
3. Na página de detalhes do cluster, escolha o botão Ações e selecione a opção Editar número de agentes.

4. Insira o número de agentes que você deseja que o cluster tenha por zona de disponibilidade. O console resume o número de agentes nas zonas de disponibilidade que serão removidos. Certifique-se de que é isso que você deseja.
5. Escolha Salvar alterações.

Para evitar a remoção acidental de um agente, o console solicita que você confirme que deseja excluir agentes.

Remova um corretor com a AWS CLI

Execute o comando a seguir, substituindo `ClusterArn` pelo nome do recurso da Amazon (ARN) que você obteve quando criou o cluster. Se você não tiver o ARN do cluster, poderá encontrá-lo listando todos os clusters. Para obter mais informações, verifique Lista de clusters do Amazon MSK. Substitua `Current-Cluster-Version` pela versão atual do cluster.

> **Important**
>
> As versões de cluster não são inteiros simples. Para encontrar a versão atual do cluster, use a DescribeClusteroperação ou o comando AWS CLI describe-cluster. Uma versão de exemplo é `KTVPDKIKX0DER`.

O *`Target-Number-of-Brokers`* parâmetro representa o número total de nós de intermediários que você deseja que o cluster tenha quando essa operação for concluída com êxito. O valor especificado *`Target-Number-of-Brokers`* deve ser um número inteiro menor que o número atual de corretores no cluster. Também deve ser um múltiplo do número de zonas de disponibilidade.

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

A saída dessa operação `update-broker-count` é semelhante ao seguinte 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
        }
    }
}
```

Nesta saída, `OperationType` é `DECREASE_BROKER_COUNT`. Se `OperationState` tiver o valor `UPDATE_IN_PROGRESS`, aguarde um pouco e execute o comando `describe-cluster-operation` novamente.

Remover um corretor com a AWS API

Para remover corretores em um cluster usando a API, consulte UpdateBrokerCounta Referência da API Amazon Managed Streaming for Apache Kafka.

Atualizar o tamanho do agente de cluster do Amazon MSK

Você pode escalar o cluster do MSK sob demanda alterando o tamanho dos agentes sem reatribuir partições do Apache Kafka. A alteração do tamanho dos agentes oferece a flexibilidade de ajustar a capacidade computacional do cluster do MSK com base nas mudanças nas workloads, sem interromper a E/S do cluster. O Amazon MSK usa o mesmo tipo de agente para todos os agentes em um determinado cluster.

Para corretores padrão, você pode atualizar o tamanho do seu agente de cluster de M5 ou T3 para M7g, T3 para M5 ou de M7g para M5.

> **Note**
>
> Você não pode migrar de uma corretora maior para uma corretora menor. Por exemplo, M7G.large a T3.small.

Para corretores Express, você pode usar apenas tamanhos de corretores M7g.

Este tópico descreve como atualizar o tamanho do broker para seu cluster MSK.

Esteja ciente de que migrar para um agente de tamanho menor pode diminuir a performance e reduzir o throughput máximo possível por agente. A migração para uma corretora maior pode aumentar o desempenho, mas pode custar mais.

A atualização do tamanho do agente ocorre de maneira contínua enquanto o cluster está ativo e em execução. Isso significa que o Amazon MSK retira um agente por vez para realizar a atualização do seu tamanho. Para obter informações sobre como tornar um cluster altamente disponível durante uma atualização de tamanho de agente, consulte the section called “Criar clusters altamente disponíveis”. Para reduzir ainda mais qualquer possível impacto sobre a produtividade, você pode realizar a atualização do tamanho do agente durante um período de baixo tráfego.

Durante uma atualização do tamanho do agente, você pode continuar produzindo e consumindo dados. No entanto, é necessário esperar até que a atualização seja concluída para poder reinicializar os agentes ou invocar qualquer uma das operações de atualização listadas nas operações do Amazon MSK.

Se você quiser atualizar o cluster para um tamanho de agente menor, recomendamos que experimente primeiro a atualização em um cluster de teste para ver como isso afetará o cenário.

> **Important**
>
> Você não poderá atualizar um cluster para um tamanho de agente menor se o número de partições por agente exceder o número máximo especificado em the section called “ Dimensione seu cluster adequadamente: número de partições por agente padrão”.

Tópicos

- Atualize o tamanho do agente de cluster Amazon MSK usando o AWS Management Console
- Atualize o tamanho do agente de cluster Amazon MSK usando o AWS CLI
- Atualizar o tamanho de agente usando a API

Atualize o tamanho do agente de cluster Amazon MSK usando o AWS Management Console

Esse processo mostra como atualizar o tamanho do agente de cluster Amazon MSK usando o AWS Management Console

1. Faça login no AWS Management Console e abra o console do Amazon MSK em https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. Escolha o cluster do MSK para o qual deseja atualizar o tamanho do agente.
3. Na página de detalhes do cluster, encontre a seção Resumo dos agentes e escolha Editar tamanho do agente.
4. Selecione o tamanho do agente desejado na lista.
5. Salve as alterações.

Atualize o tamanho do agente de cluster Amazon MSK usando o AWS CLI

Execute o comando a seguir, substituindo *ClusterArn* pelo nome do recurso da Amazon (ARN) que você obteve quando criou o cluster. Se você não tiver o ARN do cluster, poderá encontrá-lo listando todos os clusters. Para obter mais informações, consulte the section called “Listar clusters”.

1. *Current-Cluster-Version*Substitua pela versão atual do cluster e *TargetType* pelo novo tamanho que você deseja que os corretores tenham. Para saber mais sobre os tamanhos de agentes, consulte the section called “Tipos de agente”.

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

   Veja a seguir um exemplo de como usar esse 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
   ```

   A saída desse comando é semelhante ao seguinte JSON de exemplo.

   ```
   {
       "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 obter o resultado da `update-broker-type` operação, execute o comando a seguir, *`ClusterOperationArn`* substituindo-o pelo ARN obtido na saída do `update-broker-type` comando.

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

A saída desse comando `describe-cluster-operation` é semelhante ao seguinte JSON de exemplo.

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

Se `OperationState` tiver o valor `UPDATE_IN_PROGRESS`, aguarde um pouco e execute o comando `describe-cluster-operation` novamente.

Atualizar o tamanho de agente usando a API

Para atualizar o tamanho do broker usando a API, consulte UpdateBrokerType.

Você pode usar `UpdateBrokerType` para atualizar o tamanho do agende de cluster de M5 ou T3 para M7g ou de M7g para M5.

Use o LinkedIn Cruise Control para Apache Kafka com o Amazon MSK

Você pode usar LinkedIn o Cruise Control para reequilibrar seu cluster Amazon MSK, detectar e corrigir anomalias e monitorar o estado e a integridade do cluster.

Para baixar e compilar o Cruise Control

1. Crie uma EC2 instância da Amazon na mesma Amazon VPC do cluster Amazon MSK.
2. Instale o Prometheus na instância da EC2 Amazon que você criou na etapa anterior. Anote o IP privado e a porta. O número padrão da porta é 9090. Para obter informações sobre como configurar o Prometheus de modo a agregar métricas de seu cluster, consulte the section called “Monitorare com o Prometheus”.
3. Baixe o Cruise Control na EC2 instância da Amazon. (Como alternativa, você pode usar uma EC2 instância separada da Amazon para o Cruise Control, se preferir.) Para um cluster que tenha o Apache Kafka versão 2.4.*, use a versão 2.4.* mais recente do Cruise Control. Se seu cluster tiver uma versão do Apache Kafka anterior à 2.4.*, use a versão mais recente do 2.0.* Cruise Control.
4. Descompacte o arquivo do Cruise Control e acesse a pasta descompactada.
5. Execute o comando a seguir para instalar o git.

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

6. Execute o comando a seguir para inicializar o repositório local. *Your-Cruise-Control-Folder*Substitua pelo nome da sua pasta atual (a pasta que você obteve ao descompactar o download do Cruise Control).

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

7. Execute o seguinte comando para compilar o código-fonte.

```
./gradlew jar copyDependantLibs
```

Para configurar e executar o Cruise Control

1. Faça as seguintes atualizações no arquivo `config/cruisecontrol.properties`. Substitua os servidores de bootstrap de exemplo e a string bootstrap-agentes pelos valores do seu cluster.

Para obter essas strings para seu cluster, você pode ver os detalhes do cluster no console. Como alternativa, você pode usar as operações GetBootstrapBrokerse da DescribeClusterAPI ou seus equivalentes de CLI.

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

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

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

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

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

Para corretores expressos, recomendamos que você não use o `DiskCapacityGoal` em nenhuma das metas definidas nas configurações do seu analisador.

2. Edite o arquivo `config/capacityCores.json` para especificar o tamanho correto do disco, os núcleos da CPU e os limites de entrada/saída da rede. Para corretores Express, a entrada `DISK` de capacidade só é necessária para configurar o Cruise Control. Como o MSK gerencia todo o armazenamento dos corretores Express, você deve definir esse valor para um número extremamente alto, como. `Integer.MAX_VALUE (2147483647)` Para corretores padrão, você pode usar a operação de DescribeClusterAPI (ou a CLI describe-cluster) para obter o tamanho do disco. Para núcleos de CPU e limites de entrada/saída de rede, consulte Tipos de EC2 instância da Amazon.

   Standard broker config/capacityCores.json

```
{
  "brokerCapacities": [
    {
      "brokerId": "-1",
```

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

Express broker config/capacityCores.json

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

3. Opcionalmente, você pode instalar a interface do usuário do Cruise Control. Para baixá-la, acesse Como configurar o frontend do Cruise Control.
4. Execute o comando a seguir para iniciar o Cruise Control. Considere usar uma ferramenta como `screen` ou `tmux` para manter uma sessão de longa duração aberta.

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

5. Use o Cruise Control APIs ou a interface do usuário para garantir que o Cruise Control tenha os dados de carga do cluster e que esteja fazendo sugestões de rebalanceamento. A obtenção de uma janela de métricas válida pode levar alguns minutos.

> **Important**
>
> Somente as versões 2.5.60 e superiores do Cruise Control são compatíveis com corretores Express, pois os corretores Express não expõem endpoints do Zookeeper.

Usar o modelo de implantação automatizada do Cruise Control para Amazon MSK

Você também pode usar esse CloudFormation modelo para implantar facilmente o Cruise Control e o Prometheus para obter informações mais detalhadas sobre o desempenho do seu cluster Amazon MSK e otimizar a utilização de recursos.

Principais recursos:

- Provisionamento automatizado de uma EC2 instância da Amazon com Cruise Control e Prometheus pré-configurados.
- Compatibilidade com o cluster provisionado do Amazon MSK.
- Autenticação PlainText flexível com IAM.
- Nenhuma dependência do Zookeeper para o Cruise Control.
- Personalize facilmente os destinos do Prometheus, as configurações de capacidade do Cruise Control e outras configurações fornecendo seus próprios arquivos de configuração armazenados em um bucket do Amazon S3.

Diretriz de rebalanceamento de partições

Diretrizes para reatribuição de partições de Kafka

A reatribuição de partições no Kafka pode consumir muitos recursos, pois envolve a transferência de dados significativos entre corretores, o que pode causar congestionamento da rede e afetar as operações do cliente. As práticas recomendadas a seguir ajudam você a gerenciar a reatribuição de partições de forma eficaz, ajustando as taxas de aceleração, aproveitando os controles de simultaneidade e entendendo os tipos de reatribuição para minimizar a interrupção das operações do cluster.

Gerenciando a concorrência no Cruise Control

O Cruise Control fornece parâmetros de ajuste automático para controlar a simultaneidade dos movimentos de partição e liderança. Os parâmetros a seguir ajudam a manter uma carga aceitável durante as reatribuições:

- Movimentos máximos de partições simultâneas: **`num.concurrent.partition.movements.per.broker`** defina o limite máximo de movimentos de partição simultâneos entre agentes, evitando a utilização excessiva da rede.

  Example Exemplo

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

  Essa configuração limita cada corretor a mover no máximo 10 partições a qualquer momento, equilibrando a carga entre os corretores.

Use a limitação para controlar a largura de banda

- Parâmetro Throttle: Ao realizar a reatribuição de partições com`kafka-reassign-partitions.sh`, use o `--throttle parameter` para definir uma taxa máxima de transferência (em bytes por segundo) para movimentação de dados entre corretores.

  Example Exemplo

  ```
  --throttle 5000000
  ```

  Isso define uma largura de banda máxima de 5 MB/s.
- Equilibre as configurações do acelerador: Escolher uma taxa de aceleração apropriada é crucial:

  Se definido como muito baixo, a reatribuição pode levar muito mais tempo.

  Se definido como muito alto, os clientes podem experimentar aumentos de latência.
- Comece com uma taxa de aceleração conservadora e ajuste com base no monitoramento do desempenho do cluster. Teste o acelerador escolhido antes de aplicá-lo em um ambiente de produção para encontrar o equilíbrio ideal.

Teste e valide em um ambiente de teste

Antes de implementar reatribuições na produção, realize testes de carga em um ambiente de preparação com configurações semelhantes. Isso permite que você ajuste os parâmetros e minimize os impactos inesperados na produção ao vivo.

Atualizar a configuração de um cluster do Amazon MSK

Para atualizar a configuração de um cluster, certifique-se de que ele esteja no estado `ACTIVE`. Você também deve garantir que o número de partições por agente em seu cluster do MSK esteja abaixo dos limites descritos em the section called " Dimensione seu cluster adequadamente: número de partições por agente padrão". Você não pode atualizar a configuração de um cluster que exceda esses limites.

Para obter informações sobre a configuração do MSK, incluindo como criar uma configuração personalizada, quais propriedades você pode atualizar e o que acontece quando você atualiza a configuração de um cluster existente, consulte the section called "Configuração do corretor".

Tópicos

- Disponibilidade do agente durante as atualizações de configuração
- Atualizando a configuração de um cluster usando o AWS CLI
- Atualizar a configuração de um cluster do Amazon MSK usando a API

Disponibilidade do agente durante as atualizações de configuração

O Amazon MSK mantém alta disponibilidade durante a maioria das atualizações de configuração do cluster. O Amazon MSK realiza uma atualização contínua em que atualiza um corretor por vez. Durante esse processo, o cluster permanece disponível, embora os corretores individuais sejam reiniciados à medida que suas configurações forem atualizadas. No entanto, algumas alterações na configuração podem exigir que todos os corretores sejam atualizados simultaneamente, o que pode causar uma breve interrupção em todo o cluster. Para obter mais informações sobre o impacto da disponibilidade do corretor durante as atualizações, consulteConfiguração provisionada do Amazon MSK.

Antes de atualizar os clusters de produção, recomendamos que você teste suas alterações de configuração em um ambiente que não seja de produção e agende atualizações durante as janelas de manutenção.

Se você enfrentar algum problema ao atualizar seu cluster MSK, consulte Como soluciono problemas ao atualizar meu cluster Amazon MSK?

Atualizando a configuração de um cluster usando o AWS CLI

1. Copie o seguinte JSON e salve-o em um arquivo. Nomeie o arquivo `configuration-info.json`. *`ConfigurationArn`*Substitua pelo Amazon Resource Name (ARN) da configuração que você deseja usar para atualizar o cluster. A string do ARN deve estar entre aspas no seguinte JSON.

   *`Configuration-Revision`*Substitua pela revisão da configuração que você deseja usar. As revisões de configuração são inteiros (números inteiros) que começam em 1. Esse número inteiro não deve estar entre aspas no seguinte JSON.

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

2. Execute o comando a seguir, *`ClusterArn`* substituindo-o pelo ARN obtido ao criar seu cluster. Se você não tiver o ARN do cluster, poderá encontrá-lo listando todos os clusters. Para obter mais informações, consulte the section called “Listar clusters”.

   *`Path-to-Config-Info-File`*Substitua pelo caminho para seu arquivo de informações de configuração. Se você nomeou o arquivo que criou na etapa anterior `configuration-info.json` e o salvou no diretório atual, então *`Path-to-Config-Info-File`* `éconfiguration-info.json`.

   Substitua *`Current-Cluster-Version`* pela versão atual do cluster.

   > **Important**
   >
   > As versões de cluster não são inteiros simples. Para encontrar a versão atual do cluster, use a DescribeClusteroperação ou o comando AWS CLI describe-cluster. Uma versão de exemplo é `KTVPDKIKX0DER`.

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

Veja a seguir um exemplo de como usar esse 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"
```

A saída desse comando `update-cluster-configuration` é semelhante ao seguinte JSON de exemplo.

```
{
    "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 obter o resultado da `update-cluster-configuration` operação, execute o comando a seguir, *`ClusterOperationArn`* substituindo-o pelo ARN obtido na saída do `update-cluster-configuration` comando.

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

A saída desse comando `describe-cluster-operation` é semelhante ao seguinte JSON de exemplo.

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

Nesta saída, `OperationType` é `UPDATE_CLUSTER_CONFIGURATION`. Se `OperationState` tiver o valor `UPDATE_IN_PROGRESS`, aguarde um pouco e execute o comando `describe-cluster-operation` novamente.

Atualizar a configuração de um cluster do Amazon MSK usando a API

Para usar a API para atualizar a configuração de um cluster Amazon MSK, consulte UpdateClusterConfiguration.

Reinicializar um agente de um cluster do Amazon MSK

Use esta operação do Amazon MSK quando quiser reinicializar um agente para seu cluster do MSK. Para reinicializar um agente para um cluster, certifique-se de que o cluster esteja no estado `ACTIVE`.

O serviço Amazon MSK pode reinicializar os agentes do seu cluster do MSK durante a manutenção do sistema, como aplicação de patches ou atualizações de versão. A reinicialização manual de um agente permite testar a resiliência de seus clientes Kafka para determinar como eles respondem à manutenção do sistema.

Reinicialize um agente para um cluster Amazon MSK usando o AWS Management Console

Esse processo descreve como reinicializar um agente para um cluster Amazon MSK usando o. AWS Management Console

1. Abra o console do Amazon MSK em https://console.aws.amazon.com/msk/.
2. Escolha o cluster do MSK cujo agente deseja reinicializar.
3. Role para baixo até a seção Detalhes do agente e escolha o agente que deseja reinicializar.
4. Escolha o botão Reiniciar o agente.

Reinicialize um agente para um cluster Amazon MSK usando o AWS CLI

Esse processo descreve como reinicializar um agente para um cluster Amazon MSK usando o. AWS CLI

1. Execute o comando a seguir, *`ClusterArn`* substituindo-o pelo Amazon Resource Name (ARN) obtido ao criar seu cluster e pelo *`BrokerId`* ID do broker que você deseja reinicializar.

   > **Note**
   >
   > A operação `reboot-broker` só é compatível com a reinicialização de um agente por vez.

   Se você não tiver o ARN do cluster, poderá encontrá-lo listando todos os clusters. Para obter mais informações, consulte the section called “Listar clusters”.

   Se você não tiver o broker IDs para seu cluster, poderá encontrá-los listando os nós do broker. Para obter mais informações, consulte list-nodes.

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

   A saída dessa operação `reboot-broker` é semelhante ao seguinte 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 obter o resultado da `reboot-broker` operação, execute o comando a seguir, *`ClusterOperationArn`* substituindo-o pelo ARN obtido na saída do `reboot-broker` comando.

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

A saída desse comando `describe-cluster-operation` é semelhante ao seguinte JSON de exemplo.

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

Quando a operação de reinicialização estiver concluída, o `OperationState` será `REBOOT_COMPLETE`.

Reinicializar um agente de um cluster do Amazon MSK usando a API

Para reinicializar um agente em um cluster usando a API, consulte RebootBroker.

Marcar um cluster do Amazon MSK

É possível atribuir seus próprios metadados na forma de tags a um recurso do Amazon MSK, como um cluster do MSK. Uma tag é um par de chave-valor que você define para o recurso. Usar tags é uma maneira simples, porém poderosa, de gerenciar AWS recursos e organizar dados, incluindo dados de faturamento.

Tópicos

Noções básicas sobre tags para clusters do Amazon MSK

É possível usar a API do Amazon MSK para concluir as seguintes tarefas:

- Adicionar tags a um recurso do Amazon MSK.
- Listar as tags de um recurso do Amazon MSK.
- Remover as tags de um recurso do Amazon MSK.

É possível usar tags para categorizar os recursos do Amazon MSK. Por exemplo, é possível categorizar os clusters do Amazon MSK por finalidade, proprietário ou ambiente. Como você define a chave e o valor para cada marca, é possível criar um conjunto de categorias personalizado para atender às suas necessidades específicas. Por exemplo, você pode definir um conjunto de tags que ajude a monitorar os clusters por proprietário e aplicativo associado.

Estes são diversos exemplos de tags:

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

Acompanhe os custos do cluster do Amazon MSK usando a marcação

Você pode usar tags para categorizar e monitorar seus AWS custos. Quando você aplica tags aos seus AWS recursos, incluindo clusters do Amazon MSK, seu relatório de alocação de AWS custos inclui o uso e os custos agregados por tags. É possível organizar seus custos de vários serviços aplicando tags que representem categorias de negócios (como centros de custos, nomes de aplicações ou proprietários). Para obter mais informações, consulte Usar tags de alocação de custos para relatórios de faturamento personalizados no Manual do usuário do AWS Billing .

Restrições de tag

As restrições a seguir se aplicam a tags no Amazon MSK.

Restrições básicas

- O número máximo de tags por recurso é 50.
- As chaves e os valores de tags diferenciam maiúsculas de minúsculas.
- Não é possível alterar nem editar as tags de um recurso excluído.

Restrições de chaves de marcas

- Cada chave de marca deve ser exclusiva. Se uma tag for adicionada com uma chave que já estiver em uso, a nova tag substituirá o par de chave-valor existente.
- Não é possível iniciar uma chave de tag com `aws:`, pois esse prefixo é reservado para uso pela AWS. A AWS cria tags que começam com esse prefixo em seu nome, mas não é possível editá-las ou excluí-las.
- As chaves de tag devem ter entre 1 e 128 caracteres Unicode.
- As chaves de tag devem conter os seguintes caracteres: letras Unicode, dígitos, espaço em branco e os seguintes caracteres especiais: `_ . / = + - @`.

Restrições de valor das tags

- Os valores das tags devem ter entre 0 e 255 caracteres Unicode.
- Os valores das tags podem estar em branco. Caso contrário, eles devem conter os seguintes caracteres: letras Unicode, dígitos, espaço em branco e qualquer um dos seguintes caracteres especiais: `_ . / = + - @`.

Marcar recursos usando a API do Amazon MSK

É possível usar as seguintes operações para atribuir ou excluir tags de um recurso do Amazon MSK ou para listar o conjunto atual de tags de um recurso:

- ListTagsForResource
- TagResource
- UntagResource

Migrar para um cluster do Amazon MSK

O replicador do Amazon MSK pode ser usado para a migração do cluster do MSK. Consulte O que é o replicador do Amazon MSK?. Como alternativa, você pode usar o Apache MirrorMaker 2.0 para migrar de um cluster não MSK para um cluster do Amazon MSK. Para ver um exemplo de como fazer isso, consulte Migrar um cluster on-premises do Apache Kafka para o Amazon MSK usando. MirrorMaker Para obter informações sobre como usar MirrorMaker, consulte Espelhamento de dados entre clusters na documentação do Apache Kafka. Recomendamos a configuração MirrorMaker em uma configuração altamente disponível.

Um descrição das etapas a serem seguidas MirrorMaker ao usar o cluster do MSK

1. Criar o cluster de destino do MSK
2. Comece MirrorMaker com uma EC2 instância da Amazon dentro da mesma Amazon VPC como o cluster de destino.
3. Inspecione o MirrorMaker atraso.
4. Depois de MirrorMaker se atualizar, redirecione produtores e consumidores para o novo cluster usando os corretores de bootstrap do cluster MSK.
5. Encerrar MirrorMaker.

Migrar o cluster do Apache Kafka para o Amazon MSK

Suponha que você tenha um cluster do Apache Kafka chamado `CLUSTER_ONPREM`. Esse cluster é preenchido com tópicos e dados. Se quiser migrar esse cluster para um cluster recém-criado do Amazon MSK chamado `CLUSTER_AWSMSK`, esse procedimento fornecerá uma visualização de alto nível das etapas que você deverá seguir.

Para migrar o cluster existente do Apache Kafka para o Amazon MSK

1. No `CLUSTER_AWSMSK`, crie todos os tópicos que deseja migrar.

   Você não pode usar MirrorMaker essa etapa porque ela não recria automaticamente os tópicos que você deseja migrar com o nível de replicação correto. Você pode criar os tópicos no Amazon MSK com os mesmos fatores de replicação e números de partições que eles tinham em `CLUSTER_ONPREM`. Você também pode criar os tópicos com diferentes fatores de replicação e números de partições.

2. Comece MirrorMaker com uma instância que tenha acesso de leitura `CLUSTER_ONPREM` e gravação `CLUSTER_AWSMSK` a.
3. Execute o seguinte comando para espelhar todos os tópicos:

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

   Nesse comando, `config/mirrormaker-consumer.properties` aponta para um agente de bootstrap no `CLUSTER_ONPREM`; por exemplo, `bootstrap.servers=localhost:9092`. E `config/mirrormaker-producer.properties` aponta para um corretor de bootstrap em CLUSTER_AWSMSK; por exemplo,. `bootstrap.servers=10.0.0.237:9092,10.0.2.196:9092,10.0.1.233:9092`
4. Continue MirrorMaker executando em segundo plano e continue usando`CLUSTER_ONPREM`. MirrorMaker espelha todos os novos dados.
5. Verifique o andamento do espelhamento inspecionando o atraso entre o último deslocamento de cada tópico e o deslocamento atual do qual está consumindo. MirrorMaker

   Lembre-se de que MirrorMaker é simplesmente usar um consumidor e um produtor. Portanto, você pode verificar o atraso usando a ferramenta `kafka-consumer-groups.sh`. Para localizar o nome do grupo de consumidores, procure o `group.id` no arquivo `mirrormaker-consumer.properties` e use seu valor. Se essa chave não existir no arquivo, você poderá criá-la. Por exemplo, defina `group.id=mirrormaker-consumer-group`.
6. Depois de MirrorMaker terminar de espelhar todos os tópicos, pare todos os produtores e consumidores e, em seguida, pare. MirrorMaker Redirecione os produtores e consumidores para o cluster `CLUSTER_AWSMSK` alterando seus valores dos agentes de bootstrap dos produtores e consumidores. Reinicie todos os produtores e consumidores no `CLUSTER_AWSMSK`.

Migrar um cluster do Amazon MSK para outro

Você pode usar o Apache MirrorMaker 2.0 para migrar de um cluster não MSK para um cluster do MSK. Por exemplo, você pode migrar de uma versão do Apache Kafka para outra. Para ver um exemplo de como fazer isso, consulte Migrar um cluster on-premises do Apache Kafka para o Amazon MSK usando. MirrorMaker Como alternativa, o replicador do Amazon MSK pode ser usado para a migração do cluster do MSK. Para obter mais informações sobre o replicador do Amazon MSK, consulte O que é o replicador do Amazon MSK?.

MirrorMaker Práticas recomendadas 1.0

Essa lista de melhores práticas se aplica à MirrorMaker versão 1.0.

- Execute MirrorMaker no cluster de destino. Dessa forma, se ocorrer um problema de rede, as mensagens ainda estarão disponíveis no cluster de origem. Se você executa MirrorMaker no cluster de origem e os eventos são armazenados em buffer no produtor e há um problema de rede, os eventos podem ser perdidos.
- Se a criptografia for necessária em trânsito, execute-a no cluster de origem.
- Para os consumidores, defina auto.commit.enabled=false
- Para os produtores, defina
  - max.in.flight.requests.per.connection=1
  - retries=Int.Max_Value
  - acks=all
  - max.block.ms = Long.Max_Value
- Para obter um throughput alto do produtor:
  - Mensagens de buffer e lotes de mensagens de preenchimento: ajuste buffer.memory, batch.size, linger.ms
  - Ajuste os buffers de soquete: receive.buffer.bytes, send.buffer.bytes
- Para evitar a perda de dados, desative a confirmação automática na origem, para que ela MirrorMaker possa controlar as confirmações, o que normalmente acontece depois de receber o pacote do cluster de destino. Se o produtor tiver acks=all e o cluster de destino tiver min.insync.replicas definido como mais de 1, as mensagens persistirão em mais de um agente no destino antes que o consumidor confirme a compensação na origem. MirrorMaker
- Se a ordem for importante, você poderá definir novas tentativas como 0. Como alternativa, para um ambiente de produção, defina conexões máximas em trânsito como 1 para garantir que os lotes enviados não sejam confirmados fora de ordem se um lote falhar no meio. Dessa forma, cada lote enviado é repetido até que o próximo lote seja enviado. Se max.block.ms não estiver definido como o valor máximo, e se o buffer do produtor estiver cheio, poderá haver perda de dados (dependendo de algumas das outras configurações). Isso pode bloquear e retropressionar o consumidor.
- Para obter throughput alto
  - Aumente o buffer.memory.
  - Aumente o tamanho do lote.

  - Ajuste linger.ms para permitir que os lotes sejam preenchidos. Isso também permite uma melhor compactação, menos uso de largura de banda de rede e menos armazenamento no cluster. Isso resulta em maior retenção.
  - Monitore o uso da CPU e da memória.
- Para obter throughput alto do consumidor
  - Aumente o número de threads/consumidores por MirrorMaker processo: num.streams.
  - Aumente primeiro o número de MirrorMaker processos nas máquinas antes de aumentar os segmentos para permitir a alta disponibilidade.
  - Aumente o número de MirrorMaker processos primeiro na mesma máquina e depois em máquinas diferentes (com o mesmo ID de grupo).
  - Isole tópicos com throughput muito alto e use instâncias separadas MirrorMaker .
- Para gerenciamento e configuração
  - Ferramentas de gerenciamento de uso AWS CloudFormation e configuração, como Chef e Ansible.
  - Use montagens do Amazon EFS para manter todos os arquivos de configuração acessíveis em todas as EC2 instâncias da Amazon.
  - Use contêineres para facilitar o escalonamento e o gerenciamento de MirrorMaker instâncias.
- Normalmente, é preciso mais de um consumidor para saturar um produtor. MirrorMaker Portanto, configure vários consumidores. Primeiro, defina-os em diferentes máquinas para fornecer alta disponibilidade. Depois, ajuste a escala das máquinas individuais até ter um consumidor para cada partição, com consumidores distribuídos igualmente entre máquinas.
- Para obter consumo e entrega de throughput alto, ajuste os buffers de recebimento e envio porque seus padrões podem ser muito baixos. Para obter o máximo desempenho, certifique-se de que o número total de streams (num.streams) corresponda a todas as partições de tópicos que MirrorMaker estão tentando copiar para o cluster de destino.

Vantagens de MirrorMaker 2. *

- Usa a estrutura e o ecossistema do Apache Kafka Connect.
- Detecta novos tópicos e partições.
- Sincroniza automaticamente a configuração de tópicos entre clusters.
- Oferece suporte a pares de cluster “ativo/ativo”, além de qualquer número de clusters ativos.

- Fornece novas métricas, incluindo latência end-to-end de replicação em vários data centers e clusters.
- Emite deslocamentos necessários para migrar consumidores entre clusters e oferece as ferramentas para a conversão do deslocamento.
- Suporta um arquivo de configuração de alto nível para especificar vários clusters e fluxos de replicação em um só lugar, em comparação com propriedades de produtor/consumidor de baixo nível para cada processo 1.*. MirrorMaker

Excluir um cluster provisionado do Amazon MSK

> **Note**
>
> Se o cluster provisionado do Amazon MSK tiver uma política de ajuste de escala automático, recomendamos que você remova a política antes de excluí-lo. Para obter mais informações, consulte Escalabilidade automática para clusters do Amazon MSK.

Tópicos

- Exclua um cluster provisionado do Amazon MSK usando o AWS Management Console
- Exclua um cluster provisionado do Amazon MSK usando o AWS CLI
- Exclua um cluster provisionado do Amazon MSK usando a API

Exclua um cluster provisionado do Amazon MSK usando o AWS Management Console

Esse processo descreve como excluir um cluster provisionado do Amazon MSK usando o. AWS Management Console Antes de excluir um cluster do MSK, certifique-se de ter um backup de todos os dados importantes armazenados no cluster e de que não haja nenhuma tarefa programada dependente do cluster. Você não pode desfazer a exclusão de um cluster do MSK.

1. Faça login no AWS Management Console e abra o console do Amazon MSK em https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. Escolha o cluster do MSK que deseja excluir marcando a caixa de seleção ao lado dele.
3. Escolha Excluir e confirme a exclusão.

Exclua um cluster provisionado do Amazon MSK usando o AWS CLI

Esse processo descreve como excluir um cluster provisionado pelo MSK usando o. AWS CLI Antes de excluir um cluster do MSK, certifique-se de ter um backup de todos os dados importantes armazenados no cluster e de que não haja nenhuma tarefa programada dependente do cluster. Você não pode desfazer a exclusão de um cluster do MSK.

Execute o comando a seguir, substituindo *ClusterArn* pelo nome do recurso da Amazon (ARN) que você obteve quando criou o cluster. Se você não tiver o ARN do cluster, poderá encontrá-lo listando todos os clusters. Para obter mais informações, consulte the section called “Listar clusters”.

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

Exclua um cluster provisionado do Amazon MSK usando a API

A API do Amazon MSK permite que você crie e gerencie programaticamente seu cluster MSK Provisioned como parte de scripts automatizados de provisionamento ou implantação de infraestrutura. Esse processo descreve como excluir um cluster provisionado do Amazon MSK usando a API do Amazon MSK. Antes de excluir um cluster do Amazon MSK, certifique-se de ter um backup de todos os dados importantes armazenados no cluster e de que não haja nenhuma tarefa programada dependente do cluster. Você não pode desfazer a exclusão de um cluster do MSK.

Para excluir um cluster usando a API, consulte DeleteCluster.

Principais recursos e conceitos do Amazon MSK

Os clusters provisionados do Amazon MSK oferecem uma ampla variedade de recursos e capacidades para ajudá-lo a otimizar o desempenho do seu cluster e atender às suas necessidades de streaming. Os tópicos abaixo descrevem essas funcionalidades em detalhes.

- A AWS Management Console
- A Referência de API do Amazon MSK
- A Referência de comandos da CLI do Amazon MSK

Tópicos

- Gerenciamento de armazenamento para corretores padrão
- Segurança no Amazon MSK
- Configuração provisionada do Amazon MSK
- Aplicação de patches
- Agente offline e failover do cliente
- Registro em log do Amazon MSK
- Gerenciamento de metadados
- Recursos do Amazon MSK
- Versões do Apache Kafka
- Solução de problemas para o cluster do Amazon MSK

Tipos de corretores Amazon MSK

A MSK Provisioned oferece dois tipos de corretores: Standard e Express. Os corretores padrão oferecem a maior flexibilidade para configurar seus clusters, enquanto os corretores Express oferecem mais elasticidade, rendimento, resiliência e ease-of-use para executar aplicativos de streaming de alto desempenho. Consulte as subseções abaixo para obter mais detalhes sobre cada oferta. A tabela abaixo também destaca a comparação dos principais recursos entre os corretores Standard e Express.

Comparação de tipos de corretores provisionados MSK

Recurso	Corretor padrão	Corretor expresso
Gerenciamento de armazenamento	Gerenciado pelo cliente (os recursos incluem armazenamento EBS, armazenamento em camadas, taxa de transferência de armazenamento provisionado, escalabilidade automática, alertas de capacidade de armazenamento)	Totalmente gerenciado pelo MSK
Instâncias suportadas	T3, 5M, 7mg	M7g

Recurso	Corretor padrão	Corretor expresso
Considerações sobre dimensionamento e escalabilidade	Taxa de transferência, conexões, partições, armazenamento	Taxa de transferência, conexões, partições
Escalabilidade do corretor	Escala vertical e horizontal	Escala vertical e horizontal
Versões do Kafka	Consulte Versões do Apache Kafka	Começa na versão 3.6
Configuração do Apache Kafka	Mais configurável	Principalmente o MSK gerenciado para maior resiliência
Segurança	Criptografia, Private/Public acesso, autenticação e autorização - IAM, SASL/SCRAM, mTLS, texto simples, Kafka ACLs	Criptografia, Private/Public acesso, autenticação e autorização - IAM, SASL/SCRAM, mTLS, texto simples, Kafka ACLs
Monitoramento	CloudWatch, Monitoramento aberto	CloudWatch, Monitoramento aberto

> **Note**
>
> Você não pode alterar um cluster provisionado pelo MSK de um tipo de agente Standard para um tipo de agente Express trocando o tipo de agente usando a API MSK. Você precisa criar um novo cluster com o tipo de agente desejado (Standard ou Express).

Tópicos

- Corretores Amazon MSK Standard
- Corretores Amazon MSK Express

Corretores Amazon MSK Standard

Os corretores padrão do MSK Provisioned oferecem a maior flexibilidade para configurar o desempenho do seu cluster. Você pode escolher entre uma ampla variedade de configurações de cluster para obter as características de disponibilidade, durabilidade, taxa de transferência e latência necessárias para seus aplicativos. Você também pode provisionar a capacidade de armazenamento e aumentá-la conforme necessário. O Amazon MSK cuida da manutenção do hardware dos corretores padrão e dos recursos de armazenamento conectados, reparando automaticamente os problemas de hardware que possam surgir. Você pode encontrar mais detalhes neste documento sobre vários tópicos relacionados aos corretores padrão, incluindo tópicos sobre gerenciamento, configurações e manutenção de armazenamento.

Corretores Amazon MSK Express

Os corretores expressos para MSK Provisioned tornam o Apache Kafka mais simples de gerenciar, mais econômico para ser executado em grande escala e mais elástico com a baixa latência que você espera. Os corretores incluem pay-as-you-go armazenamento que se expande automaticamente e não requer dimensionamento, provisionamento ou monitoramento proativo. Dependendo do tamanho da instância selecionada, cada nó do broker pode fornecer até 3 vezes mais taxa de transferência por corretor, escalar até 20 vezes mais rápido e se recuperar 90% mais rápido em comparação com os corretores Apache Kafka padrão. Os corretores expressos vêm pré-configurados com os padrões de melhores práticas da Amazon MSK e impõem cotas de taxa de transferência do cliente para minimizar a contenção de recursos entre os clientes e as operações em segundo plano da Kafka.

Aqui estão alguns dos principais fatores e recursos a serem considerados ao usar corretores Express.

- Sem gerenciamento de armazenamento: os corretores expressos eliminam a necessidade de provisionar ou gerenciar quaisquer recursos de armazenamento. Você obtém armazenamento elástico pay-as-you-go, virtualmente ilimitado e totalmente gerenciado. Para casos de uso de alta taxa de transferência, você não precisa pensar nas interações entre instâncias de computação e volumes de armazenamento, nem sobre os gargalos de taxa de transferência associados. Esses recursos simplificam o gerenciamento de clusters e eliminam a sobrecarga operacional do gerenciamento de armazenamento.
- Escalonamento mais rápido: os corretores Express permitem que você escale seu cluster e mova partições até 20 vezes mais rápido do que nos corretores Standard. Esse recurso é crucial quando você precisa escalar seu cluster para lidar com picos de carga futuros ou escalar seu cluster para reduzir custos. Consulte as seções sobre como expandir seu cluster, remover agentes, reatribuir

partições e configurar LinkedIn o Cruise Control para rebalanceamento para obter mais detalhes sobre como escalar seu cluster.

- Maior taxa de transferência: as corretoras Express oferecem até 3 vezes mais taxa de transferência por corretora do que as corretoras Standard. Por exemplo, você pode gravar com segurança dados de até 500 MBps com cada agente Express de tamanho m7g.16xlarge, em comparação com 153,8 MBps no agente padrão equivalente (ambos os números pressupõem alocação de largura de banda suficiente para operações em segundo plano, como replicação e rebalanceamento).
- Configurado para alta resiliência: os corretores expressos oferecem automaticamente várias melhores práticas para melhorar a resiliência do seu cluster. Isso inclui barreiras em configurações críticas do Apache Kafka, cotas de produtividade e reservas de capacidade para operações em segundo plano e reparos não planejados. Esses recursos tornam mais seguro e fácil executar aplicativos Apache Kafka em grande escala. Consulte as seções sobre Configurações do Express Broker e Cota do agente Amazon MSK Express para obter mais detalhes.
- Sem janelas de manutenção: Não há janelas de manutenção para corretores Express. O Amazon MSK atualiza automaticamente o hardware do seu cluster de forma contínua. Consulte Patching for Express brokers para obter mais detalhes.

Informações adicionais sobre corretores Express

- Os corretores Express trabalham com o Apache Kafka APIs, mas ainda não oferecem suporte total à API. KStreams
- Os corretores expressos estão disponíveis apenas em uma AZs configuração 3.
- Os corretores expressos só estão disponíveis em determinados tamanhos de instância. Consulte os preços do Amazon MSK para ver a lista atualizada.
- Os corretores Express são compatíveis com as versões 3.6 e 3.8 do Apache Kafka.

Veja esses blogs

Para obter mais informações sobre os corretores MSK Express e ver um exemplo real de corretores Express em uso, leia os seguintes blogs:

- Apresentando corretores Express para Amazon MSK para oferecer alta taxa de transferência e escalabilidade mais rápida para seus clusters Kafka

- Corretores expressos para Amazon MSK: escalabilidade Kafka turbinada com desempenho até 20 vezes mais rápido

  Este blog demonstra como os corretores Express:

  - Forneça taxa de transferência mais rápida, escalabilidade rápida e melhor tempo de recuperação de falhas
  - Elimine as complexidades do gerenciamento de armazenamento

Tamanhos dos agentes do Amazon MSK

Ao criar um cluster provisionado pelo Amazon MSK, você especifica o tamanho dos corretores que deseja que ele tenha. Dependendo do tipo de corretor, o Amazon MSK oferece suporte aos seguintes tamanhos de corretor.

Tamanhos padrão de corretores

- 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

Tamanhos de corretores expressos

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

> **Note**
>
> Alguns tamanhos de corretores podem não estar disponíveis em determinadas AWS regiões. Consulte as tabelas de preços de instâncias de corretores atualizadas na página de preços do Amazon MSK para obter a lista mais recente de instâncias disponíveis por região.

Outras notas sobre tamanhos de corretores

- Os corretores M7g usam processadores AWS Graviton (processadores personalizados baseados em ARM criados pela Amazon Web Services). Os agentes M7g oferecem melhor performance de preço em relação a instâncias M5 comparáveis. Os agentes M7g consomem menos energia do que instâncias M5 comparáveis.
- O Amazon MSK oferece suporte a agentes M7g em clusters provisionados MSK executando versões 2.8.2 e 3.3.2 e superiores do Kafka.
- Os agentes M7g e M5 têm performance de throughput de linha de base superior aos agentes T3 e são recomendados para workloads de produção. Os agentes M7g e M5 também podem ter mais partições por agente do que os agentes T3. Use os agentes M7g e M5 se você estiver executando workloads de nível de produção maiores ou se exigir um número maior de partições. Para saber mais sobre os tamanhos de instância M7g e M5, consulte Instâncias de uso EC2 geral da Amazon.
- Os agentes T3 têm a capacidade de usar créditos de CPU para impulsionar temporariamente o desempenho. Use agentes T3 para desenvolvimento de baixo custo, se você estiver testando cargas de trabalho de streaming pequenas a médias ou se tiver cargas de trabalho de streaming com baixo throughput que apresentem picos temporários no throughput. Recomendamos que você faça um proof-of-concept teste para determinar se os corretores T3 são suficientes para produção ou carga de trabalho crítica. Para saber mais sobre os tamanhos de corretores T3, consulte Instâncias EC2 T3 da Amazon.

Para obter mais informações sobre como escolher tamanhos de agentes, consulte Melhores práticas para corretores Standard e Express.

Gerenciamento de armazenamento para corretores padrão

O Amazon MSK fornece recursos para ajudar você no gerenciamento do armazenamento em clusters do MSK.

> **Note**
>
> Com os corretores Express, você não precisa provisionar ou gerenciar nenhum recurso de armazenamento usado para seus dados. Isso simplifica o gerenciamento de clusters e elimina uma das causas comuns de problemas operacionais com clusters Apache Kafka.

Você também gasta menos, pois não precisa provisionar capacidade de armazenamento ociosa e paga apenas pelo que usa.

Tipo de corretor padrão

Com os corretores Standard, você pode escolher entre uma variedade de opções e recursos de armazenamento. O Amazon MSK fornece recursos para ajudar você no gerenciamento do armazenamento em clusters do MSK.

Para obter informações sobre como gerenciar a taxa de transferência, consulte???.

Tópicos

- Armazenamento hierárquico para corretores padrão
- Amplie o armazenamento de corretores Amazon MSK Standard
- Gerencie a taxa de transferência de armazenamento para corretores Standard em um cluster Amazon MSK

Armazenamento hierárquico para corretores padrão

O armazenamento em camadas é um nível de armazenamento de baixo custo para o Amazon MSK que se expande para armazenamento praticamente ilimitado, tornando econômica a criação de aplicações de streaming de dados.

Você pode criar um cluster Amazon MSK configurado com armazenamento em camadas que equilibra desempenho e custo. O Amazon MSK armazena dados de streaming no nível de armazenamento primário com desempenho otimizado até atingir os limites de retenção de tópico Apache Kafka. Em seguida, o Amazon MSK move automaticamente os dados para o novo nível de armazenamento de baixo custo.

Quando sua aplicação começa a ler dados do armazenamento em camadas, você pode esperar um aumento na latência de leitura nos primeiros bytes. Ao começar a ler os dados restantes sequencialmente do nível de baixo custo, você pode esperar latências semelhantes às do nível de armazenamento primário. Você não precisa provisionar nenhum armazenamento para o armazenamento em camadas de baixo custo nem gerenciar a infraestrutura. É possível armazenar qualquer quantidade de dados e pagar somente pelo que for usado. Esse recurso é compatível com o APIs apresentado no KIP-405: Kafka Tiered Storage.

Para obter informações sobre dimensionamento, monitoramento e otimização do seu cluster de armazenamento hierárquico MSK, consulte Melhores práticas para executar cargas de trabalho de produção usando o armazenamento hierárquico do Amazon MSK.

Veja alguns dos recursos do armazenamento em camadas:

- Você pode escalar para armazenamento praticamente ilimitado. Você não precisa adivinhar como escalar sua infraestrutura do Apache Kafka.
- Você pode reter dados por mais tempo em seus tópicos do Apache Kafka ou aumentar seu armazenamento de tópicos, sem a necessidade de aumentar o número de agentes.
- Ele fornece um buffer de segurança de maior duração para lidar com atrasos inesperados no processamento.
- Você pode reprocessar dados antigos em sua ordem de produção exata com seu código de processamento de stream existente e o Kafka APIs.
- As partições se reequilibram mais rapidamente porque os dados no armazenamento secundário não exigem replicação em discos intermediários.
- Os dados entre os agentes e o armazenamento em camadas se movem dentro da VPC e não trafegam pela Internet.
- Uma máquina cliente pode usar o mesmo processo para se conectar a novos clusters com armazenamento em camadas ativado, assim como para se conectar a um cluster sem o armazenamento em camadas ativado. Consulte Criar uma máquina cliente.

Requisitos de armazenamento em camadas de clusters do Amazon MSK

- Você deve usar a versão 3.0.0 ou superior do cliente Apache Kafka para criar um novo tópico com o armazenamento em camadas ativado. Para fazer a transição de um tópico existente para o armazenamento em camadas, você pode reconfigurar uma máquina cliente que use uma versão do cliente Kafka anterior à 3.0.0 (a versão mínima suportada do Apache Kafka é 2.8.2.) para habilitar o armazenamento em camadas. Consulte Etapa 4: criar um tópico no cluster do Amazon MSK.
- O cluster do Amazon MSK com armazenamento em camadas habilitado deve usar a versão 3.6.0 ou superior ou 2.8.2.tiered.

Restrições e limitações do armazenamento em camadas para clusters do Amazon MSK

O armazenamento em camadas tem as seguintes restrições e limitações:

- Certifique-se de que os clientes não estejam configurados como `read_committed` ao lerem de remote_tier no Amazon MSK, a menos que a aplicação esteja usando ativamente o recurso de transações.
- O armazenamento hierárquico não está disponível nas regiões AWS GovCloud (EUA).
- O armazenamento em camadas é aplicado apenas aos clusters do modo provisionado.
- O armazenamento em camadas não é compatível com o tamanho de agente t3.small.
- O período mínimo de retenção em armazenamento de baixo custo é de 3 dias. Não há período mínimo de retenção para o armazenamento primário.
- O armazenamento em camadas não oferece suporte a vários diretórios de log em um agente (recursos relacionados ao JBOD).
- O armazenamento hierárquico não é compatível com tópicos compactados. Certifique-se de que todos os tópicos com o armazenamento em camadas ativado tenham seu cleanup.policy configurado somente para “EXCLUIR”.
- O cluster de armazenamento hierárquico não oferece suporte à alteração da política log.cleanup.policy de um tópico após sua criação.
- O armazenamento hierárquico pode ser desativado para tópicos individuais, mas não para todo o cluster. Depois de desabilitado, o armazenamento em camadas não pode ser reabilitado para um tópico.
- Se você usar a versão 2.8.2.tiered do Amazon MSK, poderá migrar apenas para outra versão do Apache Kafka compatível com armazenamento em camadas. Se você não quiser continuar a usar uma versão compatível com armazenamento em camadas, crie um cluster do MSK e migre os dados para ele.
- A kafka-log-dirs ferramenta não pode relatar o tamanho dos dados de armazenamento em camadas. A ferramenta relata somente o tamanho dos segmentos de log no armazenamento primário.

Para obter informações sobre configurações e restrições padrão que você deve considerar ao configurar o armazenamento em camadas no nível do tópico, consulte. Diretrizes para a configuração de armazenamento em camadas no nível de tópico do Amazon MSK

Como os segmentos de logs são copiados para o armazenamento em camadas para um tópico do Amazon MSK

Quando você habilita o armazenamento em camadas para um tópico novo ou existente, o Apache Kafka copia segmentos de log fechados do armazenamento primário para o armazenamento em camadas.

- O Apache Kafka copia somente segmentos de log fechados. Ele copia todas as mensagens do segmento de log para o armazenamento em camadas.
- Os segmentos ativos não estão qualificados para o armazenamento em camadas. O tamanho do segmento de log (segment.bytes) ou o tempo de rolagem do segmento (segment.ms) controla a taxa de fechamento do segmento e a taxa com a qual o Apache Kafka os copia para o armazenamento em camadas.

As configurações de retenção para um tópico com o armazenamento em camadas habilitado são diferentes das configurações para um tópico sem o armazenamento em camadas habilitado. As regras a seguir controlam a retenção de mensagens em tópicos com o armazenamento em camadas habilitado:

- Você define a retenção no Apache Kafka com duas configurações: log.retention.ms (tempo) e log.retention.bytes (tamanho). Essas configurações determinam a duração total e o tamanho dos dados que o Apache Kafka retém no cluster. Independentemente de você habilitar ou não o modo de armazenamento em camadas, defina essas configurações no nível do cluster. Você pode substituir as configurações no nível do tópico pelas configurações do tópico.
- Ao habilitar o armazenamento em camadas, você também pode especificar por quanto tempo o nível primário de armazenamento de alto desempenho armazena os dados. Por exemplo, se um tópico tiver uma configuração de retenção geral (log.retention.ms) de 7 dias e retenção local (local.retention.ms) de 12 horas, o armazenamento primário do cluster vai reter os dados somente nas primeiras 12 horas. O nível de armazenamento de baixo custo retém os dados por 7 dias completos.
- As configurações usuais de retenção se aplicam ao log completo. Isso inclui suas partes primárias e em camadas.
- As configurações local.retention.ms ou local.retention.bytes controlam a retenção de mensagens no armazenamento primário. Quando os dados atingem os limites de configuração de retenção do armazenamento primário (local.retention.ms/bytes) em um log completo, o Apache Kafka copia os

dados do armazenamento primário para o armazenamento em camadas. Assim, os dados ficarão elegíveis para expiração.

- Quando o Apache Kafka copia uma mensagem em um segmento de log para o armazenamento em camadas, ele remove a mensagem do cluster com base nas configurações retention.ms ou retention.bytes.

Exemplo de cenário de armazenamento em camadas do Amazon MSK

Esse cenário ilustra como um tópico existente que tem mensagens no armazenamento primário se comporta quando o armazenamento em camadas está habilitado. Você habilita o armazenamento em camadas neste tópico ao definir remote.storage.enable como `true`. Neste exemplo, retention.ms está definido como 5 dias e local.retention.ms está definido como 2 dias. Veja a seguir a sequência de eventos quando um segmento expira.

Tempo T0: antes de você habilitar o armazenamento em camadas.

Antes de você habilitar o armazenamento em camadas para este tópico, há dois segmentos de log. Um dos segmentos está ativo para uma partição 0 de tópico existente.

Tempo T1 (< 2 dias): armazenamento em camadas habilitado. Segmento 0 copiado para o armazenamento em camadas.

Após habilitar o armazenamento em camadas para esse tópico, o Apache Kafka copia o segmento 0 de log para o armazenamento em camadas depois que o segmento satisfizer as configurações iniciais de retenção. O Apache Kafka também vai reter a cópia de armazenamento principal do segmento 0. O segmento 1 ativo ainda não está qualificado para a cópia para o armazenamento em camadas. Neste cronograma, o Amazon MSK ainda não aplica nenhuma das configurações de retenção para nenhuma das mensagens no segmento 0 e no segmento 1. (local.retenção). bytes/ms, retention.ms/bytes)

Tempo T2: retenção local em vigor.

Após 2 dias, as configurações de retenção primária entram em vigor para o segmento 0 que o Apache Kafka copiou para o armazenamento em camadas. A configuração de local.retention.ms como 2 dias determina isso. Agora, o segmento 0 expira do armazenamento primário. O segmento 1 ativo ainda não está qualificado para expiração nem está qualificado para a cópia para o armazenamento em camadas.

Tempo T3: retenção geral em vigor.

Após 5 dias, as configurações de retenção entram em vigor e o Kafka limpa o segmento 0 de log e as mensagens associadas do armazenamento em camadas. O segmento 1 ainda não está qualificado para expiração nem para cópia para armazenamento em camadas porque está ativo. O segmento 1 ainda não está fechado, portanto não é elegível para a rolagem de segmentos.

Crie um cluster Amazon MSK com armazenamento hierárquico com o AWS Management Console

Este processo descreve como criar um cluster com armazenamento em camadas do Amazon MSK usando o AWS Management Console.

1. Abra o console do Amazon MSK em https://console.aws.amazon.com/msk/.
2. Selecione Criar cluster.
3. Escolha Criação personalizada para armazenamento em camadas.
4. Especifique um nome para o cluster.
5. No Tipo de cluster, selecione Provisionado.
6. Escolha uma versão do Amazon Kafka com suporte para armazenamento em camadas a fim de que o Amazon MSK a use para criar o cluster.
7. Especifique um tamanho de agente diferente de kafka.t3.small.
8. Selecione o número de agentes que deseja que o Amazon MSK crie em cada zona de disponibilidade. O mínimo é de 1 agente por zona de disponibilidade e o máximo é de 30 agentes por cluster.
9. Especifique o número de zonas pelas quais os agentes estão distribuídos.
10. Especifique o número de agentes do Apache Kafka que estão implantados por zona.
11. Selecione Opções de armazenamento. Isso inclui Armazenamento em camadas e armazenamento do EBS para habilitar o modo de armazenamento em camadas.
12. Siga as etapas restantes no assistente de criação de cluster. Quando concluído, o Armazenamento em camadas e o armazenamento do EBS aparecerão como o modo de armazenamento de cluster na visualização Revisar e criar.
13. Selecione Create cluster (Criar cluster).

Crie um cluster Amazon MSK com armazenamento hierárquico com o AWS CLI

Para habilitar o armazenamento em camadas em um cluster, crie o cluster com a versão correta do Apache Kafka e o atributo para armazenamento em camadas. Siga o exemplo de código abaixo. Além disso, conclua as etapas da próxima seção para Crie um tópico do Kafka com o armazenamento em camadas ativado com o AWS CLI.

Consulte create-cluster para obter uma lista completa dos atributos compatíveis com a criação de clusters.

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

Crie um tópico do Kafka com o armazenamento em camadas ativado com o AWS CLI

Para concluir o processo iniciado ao criar um cluster com o armazenamento em camadas habilitado, crie também um tópico com o armazenamento em camadas habilitado com os atributos no exemplo de código adiante. Os atributos específicos para armazenamento em camadas são os seguintes:

- `local.retention.ms` (p. ex., 10 minutos) para configurações de retenção com base no tempo ou `local.retention.bytes` para limites de tamanho de segmentos de log.
- `remote.storage.enable` definido como `true` para habilitar o armazenamento em camadas.

A configuração a seguir usa local.retention.ms, mas você pode substituir esse atributo por local.retention.bytes. Esse atributo controla a quantidade de tempo que pode decorrer ou o número de bytes que o Apache Kafka pode copiar antes que o Apache Kafka copie os dados do armazenamento primário para o armazenamento em camadas. Consulte Configuração no nível de tópico para obter mais detalhes sobre os atributos de configuração compatíveis.

> **Note**
>
> Você deve usar o cliente Apache Kafka versão 3.0.0 ou superior. Essas versões são compatíveis com uma configuração chamada `remote.storage.enable` somente nas versões do cliente do `kafka-topics.sh`. Para habilitar o armazenamento em camadas em

um tópico existente usando uma versão anterior do Apache Kafka, consulte a seção Habilitar o armazenamento em camadas em um tópico existente do Amazon MSK.

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

Habilitar e desabilitar o armazenamento em camadas em um tópico existente do Amazon MSK

Estas seções abordam como habilitar e desabilitar o armazenamento em camadas em um tópico que você já criou. Para criar um novo cluster e um tópico com o armazenamento em camadas habilitado, consulte Criação de um cluster com armazenamento em camadas usando o AWS Management Console.

Habilitar o armazenamento em camadas em um tópico existente do Amazon MSK

Para habilitar armazenamento em camadas em um tópico existente, use a sintaxe de comando `alter` no seguinte exemplo. Quando você habilita o armazenamento em camadas em um tópico existente, você não está restrito a uma determinada versão do cliente 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'
```

Desabilitar o armazenamento em camadas em um tópico existente do Amazon MSK

Para desabilitar o armazenamento em camadas em um tópico existente, use a sintaxe de comando `alter` na mesma ordem em que você habilita o armazenamento em camadas.

```
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**
>
> Ao desabilitar o armazenamento em camadas, você exclui completamente os dados do tópico no armazenamento em camadas. O Apache Kafka retém os dados do

armazenamento primário, mas ainda aplica as regras de retenção primária com base em `local.retention.ms`. Após desabilitar o armazenamento em camadas em um tópico, não será possível habilitá-lo novamente. Se quiser desabilitar o armazenamento em camadas em um tópico existente, você não estará restrito a uma determinada versão do cliente Apache Kafka.

Habilite o armazenamento hierárquico em um cluster Amazon MSK existente usando a CLI AWS

> **Note**
>
> Você só pode habilitar o armazenamento em camadas se log.cleanup.policy do seu cluster estiver definido como `delete`, pois tópicos compactados não são compatíveis com o armazenamento em camadas. Posteriormente, você poderá configurar log.cleanup.policy de um tópico individual para `compact` se o armazenamento em camadas não estiver habilitado nesse tópico específico. Consulte Configuração no nível de tópico para obter mais detalhes sobre os atributos de configuração compatíveis.

1. Atualizar a versão do Kafka: as versões de cluster não são números inteiros simples. Para encontrar a versão atual do cluster, use a `DescribeCluster` operação ou o comando da `describe-cluster` AWS CLI. Uma versão de exemplo é `KTVPDKIKX0DER`.

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

2. Edite o modo de armazenamento do cluster. O exemplo de código a seguir mostra a edição do modo de armazenamento do cluster para `TIERED` usando a API `update-storage`.

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

Atualizar o armazenamento em camadas em um cluster existente do Amazon MSK usando o console

Este processo descreve como atualizar um cluster com armazenamento em camadas do Amazon MSK usando o AWS Management Console.

Certifique-se de que a versão atual do Apache Kafka do seu cluster do MSK seja 2.8.2.tiered. Consulte Atualização da versão do Apache Kafka se precisar atualizar seu cluster do MSK para a versão 2.8.2.tiered.

> **Note**
>
> Você só pode habilitar o armazenamento em camadas se log.cleanup.policy do seu cluster estiver definido como `delete`, pois tópicos compactados não são compatíveis com o armazenamento em camadas. Posteriormente, você poderá configurar log.cleanup.policy de um tópico individual para `compact` se o armazenamento em camadas não estiver habilitado nesse tópico específico. Consulte Configuração no nível de tópico para obter mais detalhes sobre os atributos de configuração compatíveis.

1. Abra o console do Amazon MSK em https://console.aws.amazon.com/msk/.
2. Acesse a página de resumo do cluster e escolha Propriedades.
3. Acesse a seção Armazenamento e escolha Editar modo de armazenamento do cluster.
4. Escolha Armazenamento em camadas e armazenamento do EBS e Salvar as alterações.

Amplie o armazenamento de corretores Amazon MSK Standard

É possível aumentar a quantidade de armazenamento do EBS por agente. Você não pode reduzir o armazenamento.

Os volumes de armazenamento permanecem disponíveis durante essa operação de expansão.

> **Important**
>
> Quando o armazenamento for escalado para um cluster do MSK, o armazenamento adicional será disponibilizado imediatamente. No entanto, o cluster requer um período de resfriamento após cada evento de escalabilidade de armazenamento. O Amazon MSK usa esse período de resfriamento para otimizar o cluster antes que ele possa ser escalado novamente. Dependendo do tamanho e da utilização do armazenamento do cluster e do tráfego, esse período pode variar de um mínimo de 6 horas a mais de 24 horas. Isso é aplicável tanto para eventos de escalonamento automático quanto para escalabilidade manual usando a UpdateBrokerStorageoperação. Para obter informações sobre como dimensionar

corretamente seu armazenamento, consulte the section called “Práticas recomendadas para agentes padrão”.

Você pode usar o armazenamento em camadas para aumentar a escala verticalmente até quantidades ilimitadas de armazenamento para seu agente. Consulte Armazenamento hierárquico para corretores padrão.

Tópicos

- Escalabilidade automática para clusters do Amazon MSK
- Dimensionamento manual para corretores padrão

Escalabilidade automática para clusters do Amazon MSK

Para expandir automaticamente o armazenamento do seu cluster em resposta ao aumento do uso, você pode configurar uma política de ajuste de escala automático de aplicações para o Amazon MSK. Em uma política de ajuste de escala automático, você define a utilização do disco de destino e a capacidade máxima de escalabilidade.

Antes de usar a escalabilidade automática para o Amazon MSK, você deve avaliar o seguinte:

- **Important**

  Uma ação de escalabilidade de armazenamento só pode ocorrer uma vez a cada 6 horas.

  Recomendamos que você comece com um volume de armazenamento do tamanho certo para suas demandas de armazenamento. Para obter orientação sobre o dimensionamento correto do seu cluster, consulte Dimensione seu cluster adequadamente: número de agentes padrão por cluster.
- O Amazon MSK não reduz o armazenamento em cluster em resposta à redução do uso. O Amazon MSK não é compatível com a redução do tamanho dos volumes de armazenamento. Se precisar reduzir o tamanho do armazenamento em cluster, você deverá migrar seu cluster existente para um cluster com armazenamento menor. Para obter informações sobre a migração de um cluster, consulte Migrar para um cluster do Amazon MSK.
- O Amazon MSK não é compatível com a redução automática da escala na horizontal nas regiões Ásia-Pacífico (Osaka) e África (Cidade do Cabo).

- Quando você associa uma política de auto-scaling ao seu cluster, o Amazon Auto EC2 Scaling cria automaticamente um alarme da CloudWatch Amazon para rastreamento de alvos. Se você excluir um cluster com uma política de auto-scaling, CloudWatch esse alarme persistirá. Para excluir o CloudWatch alarme, você deve remover uma política de auto-scaling de um cluster antes de excluir o cluster. Para saber mais sobre o rastreamento de metas, consulte Políticas de escalabilidade de rastreamento de metas para o Amazon EC2 Auto Scaling no Guia do usuário do Amazon Auto EC2 Scaling.

Tópicos

- Detalhes da política de ajuste de escala automático para o Amazon MSK
- Configurar a escalabilidade automática para o cluster do Amazon MSK

Detalhes da política de ajuste de escala automático para o Amazon MSK

Sua política de ajuste de escala automático define a seguinte métrica predefinida para seu cluster:

- Meta de utilização de armazenamento: o limite de utilização de armazenamento usado pelo Amazon MSK para acionar uma operação de ajuste de escala automático. Você pode definir a meta de utilização entre 10% e 80% da capacidade de armazenamento atual. Recomendamos que você defina a meta de utilização do armazenamento entre 50% e 60%.
- Capacidade máxima de armazenamento: o limite máximo de escalabilidade que o Amazon MSK pode definir para o armazenamento do seu agente. Você pode definir a capacidade máxima de armazenamento em até 16 TiB por agente. Para obter mais informações, consulte Cota do Amazon MSK.

Quando o Amazon MSK detecta que sua métrica `Maximum Disk Utilization` é igual ou maior que a configuração `Storage Utilization Target`, ele aumenta sua capacidade de armazenamento em um valor igual ao maior de 2 números: 10 GiB ou 10% do armazenamento atual. Por exemplo, se você tiver 1.000 GiB, esse valor será de 100 GiB. O serviço verifica a utilização do armazenamento a cada minuto. Outras operações de escalabilidade continuam aumentando o armazenamento em uma quantidade igual ao maior de 2 números: 10 GiB ou 10% do armazenamento atual.

Para determinar se ocorreram operações de auto-escalonamento, use a operação. ListClusterOperations

Configurar a escalabilidade automática para o cluster do Amazon MSK

Você pode usar o console do Amazon MSK, a API do Amazon MSK ou implementar AWS CloudFormation a escalabilidade automática para armazenamento. CloudFormation o suporte está disponível por meio de Application Auto Scaling.

> **Note**
>
> Você não pode implementar o escalabilidade automática ao criar um cluster. Primeiro, você deve criar o cluster e, em seguida, criar e habilitar uma política de ajuste de escala automático para ele. No entanto, você pode criar a política enquanto o serviço Amazon MSK cria seu cluster.

Tópicos

- Configurar a escalabilidade automática usando o AWS Management Console do Amazon MSK
- Configure o escalonamento automático usando a CLI
- Configurar a escalabilidade automática para o Amazon MSK usando a API

Configurar a escalabilidade automática usando o AWS Management Console do Amazon MSK

Este processo descreve como usar o console do Amazon MSK para implementar a escalabilidade automática para armazenamento.

1. Faça login no AWS Management Console e abra o console Amazon MSK em https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. Na lista de clusters, escolha seu cluster. Isso levará você a uma página com os detalhes sobre o cluster.
3. Na seção Ajuste de escala automático para armazenamento, escolha Configurar.
4. Crie e dê um nome a uma política de ajuste de escala automático. Especifique a meta de utilização do armazenamento, a capacidade máxima de armazenamento e a métrica de destino.
5. Selecione `Save changes`.

Quando você salvas e habilitar a nova política, ela ficará ativa para o cluster. Em seguida, o Amazon MSK expande o armazenamento do cluster quando a meta de utilização do armazenamento é atingida.

Configure o escalonamento automático usando a CLI

Este processo descreve como usar a CLI do Amazon MSK para implementar a escalabilidade automática para armazenamento.

1. Use o RegisterScalableTargetcomando para registrar um destino de utilização de armazenamento.
2. Use o PutScalingPolicycomando para criar uma política de expansão automática.

Configurar a escalabilidade automática para o Amazon MSK usando a API

Este processo descreve como usar a API do Amazon MSK para implementar a escalabilidade automática para armazenamento.

1. Use a RegisterScalableTargetAPI para registrar uma meta de utilização de armazenamento.
2. Use a PutScalingPolicyAPI para criar uma política de expansão automática.

Dimensionamento manual para corretores padrão

Para aumentar o armazenamento, aguarde que o cluster esteja no estado `ACTIVE`. O escalonamento de armazenamento tem um período de resfriamento de pelo menos 6 horas entre os eventos. Embora a operação disponibilize armazenamento adicional imediatamente, o serviço realiza otimizações em seu cluster que podem levar até 24 horas ou mais. A duração dessas otimizações é proporcional ao tamanho do seu armazenamento.

Ampliando o armazenamento do corretor usando o AWS Management Console

1. Abra o console do Amazon MSK em https://console.aws.amazon.com/msk/.
2. Escolha o cluster do MSK para o qual deseja atualizar o armazenamento do agente.
3. Na seção Armazenamento, escolha Editar.
4. Especifique o volume de armazenamento desejado. Só é possível aumentar a quantidade de armazenamento, não é possível reduzi-la.
5. Escolha Salvar alterações.

Ampliando o armazenamento do corretor usando o AWS CLI

Execute o comando a seguir, substituindo *ClusterArn* pelo nome do recurso da Amazon (ARN) que você obteve quando criou o cluster. Se você não tiver o ARN do cluster, poderá encontrá-lo listando todos os clusters. Para obter mais informações, consulte the section called “Listar clusters”.

Substitua *Current-Cluster-Version* pela versão atual do cluster.

> **Important**
>
> As versões de cluster não são inteiros simples. Para encontrar a versão atual do cluster, use a DescribeClusteroperação ou o comando AWS CLI describe-cluster. Uma versão de exemplo é `KTVPDKIKX0DER`.

O *Target-Volume-in-GiB* parâmetro representa a quantidade de armazenamento que você deseja que cada corretor tenha. Só é possível atualizar o armazenamento de todos os agentes. Não é possível especificar agentes individuais dos quais atualizar o armazenamento. O valor especificado *Target-Volume-in-GiB* deve ser um número inteiro maior que 100 GiB. O armazenamento por agente após a operação de atualização não pode exceder 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}'
```

Aumentar a escala verticalmente do armazenamento do agente usando a API

Para atualizar o armazenamento de um broker usando a API, consulte UpdateBrokerStorage.

Gerencie a taxa de transferência de armazenamento para corretores Standard em um cluster Amazon MSK

Para obter informações sobre como provisionar a taxa de transferência usando o console, a CLI e a API do Amazon MSK, consulte. ???

Tópicos

- Configurações de throughput máximo e gargalos de throughput de agentes do Amazon MSK
- Avalie o throughput de armazenamento de um cluster do Amazon MSK

- Valores de atualização de configuração para armazenamento provisionado em um cluster do Amazon MSK
- Provisione a taxa de transferência de armazenamento para corretores padrão em um cluster Amazon MSK

Configurações de throughput máximo e gargalos de throughput de agentes do Amazon MSK

Há várias causas de gargalos na taxa de transferência do corretor: taxa de transferência de volume, taxa de transferência da rede Amazon para EC2 Amazon EBS e taxa de transferência de saída da Amazon. EC2 Você pode ativar o throughput do armazenamento provisionado para ajustar o throughput do volume. No entanto, as limitações de taxa de transferência do corretor podem ser causadas pela taxa de transferência da rede Amazon EC2 para Amazon EBS e pela taxa de transferência de saída da Amazon EC2.

A taxa de EC2 saída da Amazon é afetada pelo número de grupos de consumidores e consumidores por grupo de consumidores. Além disso, a taxa de transferência da rede Amazon EC2 para Amazon EBS e a taxa de transferência de EC2 saída da Amazon são maiores para corretoras maiores.

Para volumes com tamanhos de 10 GiB ou mais, você pode provisionar um throughput de armazenamento de 250 MiB por segundo ou mais. O valor de 250 MiB por segundo é o padrão. Para provisionar o throughput de armazenamento, você deve escolher o tamanho de agente kafka.m5.4xlarge ou maior (ou kafka.m7g.2xlarge ou maior), e você pode especificar o throughput máximo conforme apresentado na tabela a seguir.

tamanho do agente	Throughput máximo de armazenamento (MiB/segundo)
kafka.m5.4xlarge	593
kafka.m5.8xlarge	850
kafka.m5.12xlarge	1000
kafka.m5.16xlarge	1000
kafka.m5.24xlarge	1000
kafka.m7g.2xlarge	312,5

tamanho do agente	Throughput máximo de armazenamento (MiB/segundo)
kafka.m7g.4xlarge	625
kafka.m7g.8xlarge	1000
kafka.m7g.12xlarge	1000
kafka.m7g.16xlarge	1000

Avalie o throughput de armazenamento de um cluster do Amazon MSK

Você pode usar as métricas `VolumeReadBytes` e `VolumeWriteBytes` para medir o throughput médio de armazenamento de um cluster. A soma dessas duas métricas fornece o throughput médio de armazenamento em bytes. Para obter o throughput médio de armazenamento de um cluster, defina essas duas métricas como SUM e o período como 1 minuto e então aplique a fórmula a seguir.

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

Para obter mais informações sobre as métricas `VolumeReadBytes` e `VolumeWriteBytes`, consulte the section called “Monitoramento no nível PER_BROKER”.

Valores de atualização de configuração para armazenamento provisionado em um cluster do Amazon MSK

Você pode atualizar sua configuração do Amazon MSK antes ou depois de ativar o throughput provisionado. No entanto, você não verá o throughput desejado até realizar estas duas ações: atualizar o parâmetro de configuração `num.replica.fetchers` e ativar o throughput provisionado.

Na configuração padrão do Amazon MSK, `num.replica.fetchers` tem um valor de 2. Para atualizar seu `num.replica.fetchers`, você pode usar os valores sugeridos na tabela a seguir. Estes valores são para fins de orientação. Recomendamos ajustar os valores com base no seu caso de uso.

tamanho do agente	num.replica.fetchers
kafka.m5.4xlarge	4
kafka.m5.8xlarge	8
kafka.m5.12xlarge	14
kafka.m5.16xlarge	16
kafka.m5.24xlarge	16

Sua configuração atualizada pode não entrar em vigor por até 24 horas e isso pode levar mais tempo quando um volume de origem não for totalmente utilizado. No entanto, o desempenho do volume de transição é, no mínimo, igual ao desempenho dos volumes de armazenamento de origem durante o período de migração. Um volume de 1 TiB totalmente utilizado normalmente leva aproximadamente 6 horas para migrar para uma configuração atualizada.

Provisione a taxa de transferência de armazenamento para corretores padrão em um cluster Amazon MSK

Os agentes do Amazon MSK mantêm os dados em volumes de armazenamento. I/O O armazenamento é consumido quando os produtores gravam no cluster, quando os dados são replicados entre corretores e quando os consumidores leem dados que não estão na memória. O throughput de armazenamento em volume é a taxa na qual os dados podem ser gravados e lidos em um volume de armazenamento. O throughput de armazenamento provisionado é a capacidade de especificar essa taxa para os agentes em seu cluster.

Você pode especificar a taxa de throughput provisionado em MiB por segundo para clusters cujos agentes sejam do tamanho `kafka.m5.4xlarge` ou maiores e se o volume de armazenamento for de 10 GiB ou mais. É possível especificar o throughput provisionado durante a criação do cluster. Você também pode ativar ou desativar o throughput provisionado para um cluster que esteja no estado `ACTIVE`.

Para obter informações sobre como gerenciar a taxa de transferência, consulte???.

Tópicos

- Provisione a taxa de transferência de armazenamento em cluster do Amazon MSK usando o AWS Management Console

- Provisione a taxa de transferência de armazenamento em cluster do Amazon MSK usando o AWS CLI
- Provisionar o throughput de armazenamento ao criar um cluster do Amazon MSK usando a API

Provisione a taxa de transferência de armazenamento em cluster do Amazon MSK usando o AWS Management Console

Esse processo mostra um exemplo de como você pode usar o AWS Management Console para criar um cluster Amazon MSK com a taxa de transferência provisionada ativada.

1. Faça login no AWS Management Console e abra o console Amazon MSK em https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. Selecione Criar cluster.
3. Escolha Criação personalizada.
4. Especifique um nome para o cluster.
5. Na seção Armazenamento, escolha Habilitar.
6. Escolha um valor para o throughput de armazenamento por agente.
7. Selecione uma VPC, as zonas e sub-redes, além dos grupos de segurança.
8. Escolha Próximo.
9. Na parte inferior da etapa Segurança, escolha Avançar.
10. Na parte inferior da etapa Monitoramento e tags, escolha Avançar.
11. Revise as configurações do cluster e escolha Criar cluster.

Provisione a taxa de transferência de armazenamento em cluster do Amazon MSK usando o AWS CLI

Esse processo mostra um exemplo de como você pode usar o AWS CLI para criar um cluster com a taxa de transferência provisionada ativada.

1. Copie e cole o JSON a seguir em um arquivo. Substitua os espaços reservados do ID da sub-rede IDs e do grupo de segurança pelos valores da sua conta. Nomeie e salve o arquivo como `cluster-creation.json`.

```
{
```

```
    "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. Execute o AWS CLI comando a seguir no diretório em que você salvou o arquivo JSON na etapa anterior.

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

Provisionar o throughput de armazenamento ao criar um cluster do Amazon MSK usando a API

Para configurar a taxa de transferência de armazenamento provisionado ao criar um cluster, use a V2. CreateCluster

Segurança no Amazon MSK

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de uma arquitetura de data center e rede criada para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O modelo de responsabilidade compartilhada descreve isso como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de AWS de . Para saber mais sobre os programas de conformidade aplicáveis ao Amazon Managed Streaming for Apache Kafka, consulte Serviços da Amazon Web Services no escopo por programa de conformidade.
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da sua empresa e as leis e normas aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Amazon MSK. Os tópicos a seguir mostram como configurar o Amazon MSK para atender aos seus objetivos de segurança e compatibilidade. Saiba também como usar outros serviços da Amazon Web Services que ajudam você a monitorar e proteger seus recursos do Amazon MSK.

Tópicos

Proteção de dados no Amazon Managed Streaming for Apache Kafka

O modelo de responsabilidade AWS compartilhada O modelo se aplica à proteção de dados no Amazon Managed Streaming for Apache Kafka. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre a privacidade de dados, consulte as Data Privacy FAQ. Para obter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and RGPD no Blog de segurança da AWS .

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como trabalhar com CloudTrail trilhas no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para obter mais informações sobre os endpoints FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-3.

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sigilosas, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com o Amazon MSK ou outro Serviços da AWS usando o console, a API ou AWS SDKs. AWS CLI Quaisquer dados inseridos em tags ou em campos de texto de formato livre

usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é fortemente recomendável que não sejam incluídas informações de credenciais no URL para validar a solicitação nesse servidor.

Tópicos

- Criptografia do Amazon MSK
- Conceitos básicos sobre criptografia do Amazon MSK
- Use o Amazon MSK APIs com endpoints de interface VPC

Criptografia do Amazon MSK

O Amazon MSK fornece opções de criptografia de dados que você pode usar para atender a requisitos rigorosos de gerenciamento de dados. É necessário renovar a cada 13 meses os certificados que o Amazon MSK usa para criptografia. O Amazon MSK renova automaticamente esses certificados para todos os clusters. Os clusters de agentes expressos permanecem no `ACTIVE` estado quando o Amazon MSK inicia a operação de atualização do certificado. Para clusters de corretores padrão, o Amazon MSK define o estado do cluster para `MAINTENANCE` quando ele inicia a operação de atualização do certificado. O MSK o redefine para `ACTIVE` quando a atualização for concluída. Enquanto um cluster está na operação de atualização do certificado, você pode continuar produzindo e consumindo dados, mas não pode realizar nenhuma operação de atualização nele.

Criptografia do Amazon MSK em repouso

O Amazon MSK se integra ao AWS Key Management Service (KMS) para oferecer uma criptografia transparente no lado do servidor. O Amazon MSK sempre criptografa seus dados em repouso. Ao criar um cluster do MSK, você pode especificar a AWS KMS key que deseja que o Amazon MSK use para criptografar seus dados em repouso. Se você não especificar uma chave do KMS, o Amazon MSK criará uma Chave gerenciada pela AWS para você e a usará em seu nome. Para ter mais informações sobre as chaves do KMS, consulte AWS KMS keys no Guia do desenvolvedor do AWS Key Management Service .

Criptografia do Amazon MSK em trânsito

O Amazon MSK usa TLS 1.2. Por padrão, ele criptografa os dados em trânsito entre os agentes do seu cluster do MSK. É possível substituir esse padrão no momento de criação do cluster.

Para a comunicação entre clientes e agentes, é necessário especificar uma destas três configurações:

- Permitir somente dados criptografados por TLS. Essa é a configuração padrão.
- Permitir dados não criptografados e dados criptografados por TLS.
- Permitir apenas dados não criptografados.

Os corretores do Amazon MSK usam certificados públicos AWS Certificate Manager . Portanto, qualquer armazenamento confiável que confie no Amazon Trust Services também confia nos agentes do Amazon MSK.

Embora seja altamente recomendável habilitar a criptografia em trânsito, isso pode acrescentar sobrecarga à CPU e alguns milissegundos de latência. Contudo, a maioria dos casos de uso não é afetada por essas diferenças, e a magnitude do impacto depende da configuração do cluster, dos clientes e do perfil de uso.

Conceitos básicos sobre criptografia do Amazon MSK

Ao criar um cluster do MSK, você pode especificar configurações de criptografia no formato JSON. Veja um exemplo a seguir.

```
{
   "EncryptionAtRest": {
       "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/abcdabcd-1234-
abcd-1234-abcd123e8e8e"
    },
   "EncryptionInTransit": {
        "InCluster": true,
        "ClientBroker": "TLS"
    }
}
```

Para `DataVolumeKMSKeyId`, é possível especificar uma chave gerenciada pelo cliente ou a Chave gerenciada pela AWS para o MSK na sua conta (`alias/aws/kafka`). Se você não especificar`EncryptionAtRest`, o Amazon MSK ainda criptografa seus dados em repouso sob o. Chave gerenciada pela AWS Para determinar qual chave o cluster está usando, envie uma solicitação `GET` ou invoque a operação de API `DescribeCluster`.

Para `EncryptionInTransit`, o valor padrão de `InCluster` é verdadeiro, mas será possível defini-lo como falso se não quiser que o Amazon MSK criptografe seus dados conforme eles passam pelos agentes.

Para especificar o modo de criptografia para dados em trânsito entre clientes e agentes, defina `ClientBroker` como um dos três valores: `TLS`, `TLS_PLAINTEXT` ou `PLAINTEXT`.

Tópicos

- Especificar as configurações de criptografia ao criar um cluster do Amazon MSK
- Testar a criptografia TLS do Amazon MSK

Especificar as configurações de criptografia ao criar um cluster do Amazon MSK

Este processo descreve como especificar as configurações de criptografia ao criar um cluster do Amazon MSK.

Especificar as configurações de criptografia ao criar um cluster

1. Salve o conteúdo do exemplo anterior em um arquivo e dê ao arquivo qualquer nome que desejar. Por exemplo, nomeie-o como `encryption-settings.json`.
2. Execute o comando `create-cluster` e use a opção `encryption-info` para apontar para o arquivo onde você salvou a configuração JSON. Veja um exemplo a seguir. *`{YOUR MSK VERSION}`*Substitua por uma versão que corresponda à versão do cliente Apache Kafka. Para obter informações sobre como encontrar a versão de cluster do MSK, consulte Determinando a versão do cluster MSK. Esteja ciente de que usar uma versão do cliente Apache Kafka que não seja igual à sua versão de cluster do MSK pode resultar em corrupção, perda e tempo de inatividade dos dados do 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
```

Veja a seguir um exemplo de uma resposta bem-sucedida após a execução desse comando.

```
{
    "ClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/SecondTLSTest/
abcdabcd-1234-abcd-1234-abcd123e8e8e",
    "ClusterName": "ExampleClusterName",
    "State": "CREATING"
}
```

Testar a criptografia TLS do Amazon MSK

Este processo descreve como testar a criptografia TLS no Amazon MSK.

Como testar a criptografia por TLS

1. Crie uma máquina de cliente seguindo as orientações em the section called “Criar uma máquina cliente”.
2. Instale o Apache Kafka na máquina de cliente.
3. Neste exemplo, o armazenamento confiável da JVM para se comunicar com o cluster do MSK. Para fazer isso, crie primeiramente uma pasta chamada `/tmp` na máquina cliente. Depois, acesse a pasta `bin` da instalação do Apache Kafka e execute o comando a seguir. (Seu caminho da JVM pode 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. Enquanto ainda estiver na pasta `bin` da instalação do Apache Kafka na máquina cliente, crie um arquivo de texto chamado `client.properties` com o conteúdo a seguir.

   ```
   security.protocol=SSL
   ssl.truststore.location=/tmp/kafka.client.truststore.jks
   ```

5. Execute o comando a seguir em uma máquina que tenha o AWS CLI instalado, *`clusterARN`* substituindo-o pelo ARN do seu cluster.

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

   Um resultado bem-sucedido tem a aparência a seguir. Salve este resultado porque você precisará dele na próxima etapa.

   ```
   {
       "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. Execute o comando a seguir, *`BootstrapBrokerStringTls`* substituindo-o por um dos endpoints do broker que você obteve na etapa anterior.

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

7. Abra uma nova janela de comando e conecte-se à mesma máquina cliente. Depois, execute o comando a seguir para criar um consumidor de console.

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBrokerStringTls --consumer.config client.properties --topic
 TLSTestTopic
```

8. Na janela do produtor, digite uma mensagem de texto seguida de um retorno e procure a mesma mensagem na janela do consumidor. O Amazon MSK criptografou essa mensagem em trânsito.

Para obter mais informações sobre como configurar clientes do Apache Kafka para trabalhar com dados criptografados, consulte Configurar clientes do Kafka.

Use o Amazon MSK APIs com endpoints de interface VPC

Você pode usar uma interface VPC Endpoint, alimentada por AWS PrivateLink, para evitar que o tráfego entre sua Amazon VPC e Amazon MSK saia APIs da rede Amazon. Os VPC Endpoints de interface não exigem um gateway de internet, dispositivo NAT, conexão VPN ou conexão Direct AWS Connect. AWS PrivateLinké uma AWS tecnologia que permite a comunicação privada entre AWS serviços usando uma interface de rede elástica com privacidade IPs em sua Amazon VPC. Para obter mais informações, consulte Amazon Virtual Private Cloud e Interface VPC Endpoints ().AWS PrivateLink

Seus aplicativos podem se conectar ao Amazon MSK Provisioned e ao MSK Connect usando. APIs AWS PrivateLink Para começar, crie uma interface VPC Endpoint para sua API Amazon MSK para iniciar o fluxo de tráfego de e para seus recursos da Amazon VPC por meio da interface VPC Endpoint. Os endpoints VPC de interface habilitada para FIPS estão disponíveis para as regiões dos EUA. Para obter mais informações, consulte Criar um endpoint de interface.

Usando esse recurso, seus clientes Apache Kafka podem buscar dinamicamente as cadeias de conexão para se conectar aos recursos do MSK Provisioned ou do MSK Connect sem percorrer a Internet para recuperar as cadeias de conexão.

Ao criar um endpoint VPC de interface, escolha um dos seguintes endpoints de nome de serviço:

Para MSK Provisioned:

- com.amazonaws.region.kafka
- com.amazonaws.region.kafka-fips (habilitado para FIPS)

Onde região é o nome da sua região. Escolha esse nome de serviço para trabalhar com o MSK APIs Provisioned compatível. Para obter mais informações, consulte Operações na referência https://docs.aws.amazon.com/msk/ 1.0/api/.

Para o MSK Connect:

- com.amazonaws.region.kafkaconnect

Onde região é o nome da sua região. Escolha esse nome de serviço para trabalhar com o MSK Connect APIs compatível. Para obter mais informações, consulte Ações na referência da API Amazon MSK Connect.

Para obter mais informações, incluindo step-by-step instruções para criar um endpoint VPC de interface, consulte Criação de um endpoint de interface no Guia.AWS PrivateLink

Controle o acesso aos VPC endpoints para Amazon MSK Provisioned ou MSK Connect APIs

As políticas de VPC endpoint permitem controlar o acesso anexando uma política a um VPC endpoint ou usando campos adicionais em uma política anexada a um usuário, grupo ou função do IAM para restringir o acesso a ocorrer somente por meio do VPC endpoint especificado. Use o exemplo de política apropriado para definir as permissões de acesso para o serviço MSK Provisioned ou MSK Connect.

Se você não associar uma política ao criar um endpoint, a Amazon VPC associará uma política padrão que permita o acesso total ao serviço. Uma política de endpoint não substitui as políticas do IAM nem as políticas fundamentadas na identidade e específicas do serviço. É uma política separada para controlar o acesso do endpoint ao serviço especificado.

Para obter mais informações, consulte Como controlar o acesso aos serviços com VPC Endpoints no Guia.AWS PrivateLink

MSK Provisioned — VPC policy example

Acesso somente leitura.

Esse exemplo de política pode ser anexado a um VPC endpoint. (Para obter mais informações, consulte Como controlar o acesso aos recursos da Amazon VPC). Ele restringe as ações a apenas listar e descrever as operações por meio do VPC endpoint ao qual está anexado.

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

MSK Provisioned — exemplo de política de endpoint de VPC

Restringir o acesso a um cluster MSK específico

Esse exemplo de política pode ser anexado a um VPC endpoint. Ele restringe o acesso a um cluster Kafka específico por meio do VPC endpoint ao qual está conectado.

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

MSK Connect — VPC endpoint policy example

Listar conectores e criar um novo conector

Veja a seguir um exemplo de uma política de endpoint para o MSK Connect. Essa política permite que a função especificada liste conectores e crie um novo conector.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "MSKConnectPermissions",
            "Effect": "Allow",
            "Action": [
                "kafkaconnect:ListConnectors",
                "kafkaconnect:CreateConnector"
            ],
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:role/<ExampleRole>"
                ]
            }
        }
    ]
}
```

MSK Connect — exemplo de política de endpoint de VPC

Permite somente solicitações de um endereço IP específico na VPC especificada

O exemplo a seguir mostra uma política que só permite a efetivação de solicitações provenientes de um endereço IP especificado na VPC estabelecida. Solicitações de outros endereços IP não são aceitas.

```
{
    "Statement": [
        {
            "Action": "kafkaconnect:*",
            "Effect": "Allow",
            "Principal": "*",
            "Resource": "*",
            "Condition": {
```

```
                "IpAddress": {
                    "aws:VpcSourceIp": "192.0.2.123"
                },
        "StringEquals": {
                    "aws:SourceVpc": "vpc-555555555555"
                }
            }
        }
    ]
}
```

Autenticação e autorização para Amazon MSK APIs

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (ter permissões) para utilizar os recursos do Amazon MSK. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

- Como o Amazon MSK funciona com o IAM
- Exemplos de política baseada em identidade do Amazon MSK
- Perfis vinculados ao serviço para o Amazon MSK
- AWS políticas gerenciadas para o Amazon MSK
- Solução de problemas de identidade e acesso do Amazon MKS

Como o Amazon MSK funciona com o IAM

Antes de usar o IAM para gerenciar o acesso ao Amazon MSK, você deve entender quais recursos do IAM estão disponíveis para uso com o Amazon MSK. Para obter uma visão de alto nível de como o Amazon MSK e outros AWS serviços funcionam com o IAM, consulte AWS Serviços que funcionam com o IAM no Guia do usuário do IAM.

Tópicos

- Políticas baseadas em identidade do Amazon MSK
- Políticas baseadas em recurso do Amazon MSK
- Autorização baseada em tags do Amazon MSK

- Perfis do IAM para o Amazon MSK

Políticas baseadas em identidade do Amazon MSK

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. O Amazon MSK é compatível com ações, chaves de condição e recursos específicos. Para conhecer todos os elementos usados em uma política JSON, consulte Referência de elementos de política JSON do IAM no Guia do usuário do IAM.

Ações para políticas baseadas em identidade do Amazon MSK

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da operação de AWS API associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

As ações de política no Amazon MSK usam o seguinte prefixo antes da ação: `kafka:`. Por exemplo, para conceder permissão a alguém para descrever um cluster do MSK com a operação de API `DescribeCluster` do Amazon MSK, inclua a ação `kafka:DescribeCluster` na política. As instruções de política devem incluir um elemento `Action` ou `NotAction`. O Amazon MSK define seu próprio conjunto de ações que descrevem as tarefas que você pode executar com esse serviço.

Para especificar várias ações em uma única instrução, separe-as com vírgulas, como segue:

```
"Action": ["kafka:action1", "kafka:action2"]
```

Você também pode especificar várias ações usando caracteres curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `Describe`, inclua a seguinte ação:

```
"Action": "kafka:Describe*"
```

Para ver uma lista de ações do Amazon MSK, consulte Ações, recursos e chaves de condição do Amazon Managed Streaming for Apache Kafka no Guia do usuário do IAM.

Recursos para políticas baseadas em identidade do Amazon MSK

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou `NotResource`. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

O recurso de instância do Amazon MSK tem o seguinte ARN:

```
arn:${Partition}:kafka:${Region}:${Account}:cluster/${ClusterName}/${UUID}
```

Para obter mais informações sobre o formato de ARNs, consulte Amazon Resource Names (ARNs) e AWS Service Namespaces.

Por exemplo, para especificar a instância `CustomerMessages` na instrução, use o seguinte ARN:

```
"Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/CustomerMessages/abcd1234-
abcd-dcba-4321-a1b2abcd9f9f-2"
```

Para especificar todas as instâncias que pertencem a uma conta específica, use o caractere curinga (*):

```
"Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/*"
```

Algumas ações do Amazon MSK, como as usadas para a criação de recursos, não podem ser executadas em um recurso específico. Nesses casos, você deve utilizar o caractere curinga (*).

```
"Resource": "*"
```

Para especificar vários recursos em uma única instrução, separe-os ARNs com vírgulas.

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

Para ver uma lista dos tipos de recursos do Amazon MSK e seus ARNs, consulte Resources Defined by Amazon Managed Streaming for Apache Kafka no Guia do usuário do IAM. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte Ações definidas pelo Amazon Managed Streaming for Apache Kafka.

Chaves de condição para políticas baseadas em identidade do Amazon MSK

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usem agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos de `Condition` em uma declaração ou várias chaves em um único elemento de `Condition`, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma `OR` operação lógica. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um recurso somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte Elementos da política do IAM: variáveis e tags no Guia do usuário do IAM.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia do usuário do IAM.

O Amazon MSK define seu próprio conjunto de chaves de condição e também é compatível com o uso de algumas chaves de condição globais. Para ver todas as chaves de condição AWS globais, consulte Chaves de contexto de condição AWS global no Guia do usuário do IAM.

Para ver uma lista das chaves de condição do Amazon MSK, consulte Chaves de condição para o Amazon Managed Streaming for Apache Kafka no Guia do usuário do IAM. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte Ações definidas pelo Amazon Managed Streaming for Apache Kafka.

Exemplos de políticas baseadas em identidade do Amazon MSK

Para visualizar exemplos de políticas baseadas em identidade do Amazon MSK, consulte Exemplos de política baseada em identidade do Amazon MSK.

Políticas baseadas em recurso do Amazon MSK

O Amazon MSK é compatível com uma política de cluster (também conhecida como política baseada em recurso) para uso com clusters do Amazon MSK. Você pode usar uma política de cluster para definir quais entidades principais do IAM têm permissões entre contas para configurar a conectividade privada com seu cluster do Amazon MSK. Quando usada com a autenticação de cliente do IAM, você também pode usar a política de cluster para definir de modo granular as permissões do plano de dados do Kafka para os clientes conectados.

Para ver um exemplo de como configurar uma política de cluster, consulte Etapa 2: anexar uma política de cluster ao cluster do MSK.

Autorização baseada em tags do Amazon MSK

É possível anexar tags a clusters do Amazon MSK. Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `kafka:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* ou chaves de condição `aws:TagKeys`. Para obter informações sobre a marcação de recursos do Amazon MSK, consulte. the section called “Marcar um cluster”

Você só pode controlar o acesso ao cluster com a ajuda de tags. Para marcar tópicos e grupos de consumidores, você precisa adicionar uma declaração separada em suas políticas sem tags.

Para ver um exemplo de uma política baseada em identidade para limitar o acesso a um cluster com base nas tags desse cluster, consulte. Como acessar clusters do Amazon MSK com base em tags

Você pode usar condições em sua política baseada em identidade para controlar o acesso aos recursos do Amazon MSK com base em tags. O exemplo a seguir mostra uma política que permite

ao usuário descrever o cluster, obter seus corretores de bootstrap, listar seus nós de agente, atualizá-lo e excluí-lo. No entanto, essa política concede permissão somente se a tag do cluster `Owner` tiver o valor desse usuário`username`. A segunda declaração na política a seguir permite acesso aos tópicos do cluster. A primeira declaração nesta política não autoriza o acesso a nenhum tópico.

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/*"
      ]
    }
  ]
}
```

Perfis do IAM para o Amazon MSK

Um perfil do IAM é uma entidade dentro da sua conta da Amazon Web Services que tem permissões específicas.

Usar credenciais temporárias com o Amazon MSK

É possível usar credenciais temporárias para fazer login com federação, assumir um perfil do IAM ou assumir um perfil entre contas. Você obtém credenciais de segurança temporárias chamando operações de AWS STS API, como AssumeRoleou GetFederationToken.

A Amazon MSK é compatível com o uso de credenciais temporárias.

Perfis vinculados a serviço

Os perfis vinculados a serviço permitem que os serviços da Amazon Web Services acessem recursos em outros serviços para concluir uma ação em seu nome. Os perfis vinculados a serviço aparecem em sua conta do IAM e são de propriedade do serviço. Um administrador do pode visualizar, mas não pode editar as permissões para funções vinculadas ao serviço.

O Amazon ECS MSK é compatível com perfis vinculados a serviço. Para obter detalhes sobre como criar ou gerenciar perfis vinculados a serviço do Amazon MSK, consulte the section called “Perfis vinculados a serviço”.

Exemplos de política baseada em identidade do Amazon MSK

Por padrão, usuários e perfis do IAM não têm permissão para executar ações de API do Amazon MSK. Um administrador deve criar as políticas do IAM que concedam aos usuários e aos perfis permissões para executar operações de API específicas nos recursos especificados que precisam. O administrador deve anexar essas políticas aos usuários ou grupos do IAM que exigem essas permissões.

Para saber como criar uma política baseada em identidade do IAM usando esses exemplos de documentos de política JSON, consulte Criar políticas na guia JSON no Guia do usuário do IAM.

Tópicos

- Práticas recomendadas de política
- Permitir que os usuários visualizem suas próprias permissões
- Acessar um cluster do Amazon MSK

- Como acessar clusters do Amazon MSK com base em tags

Práticas recomendadas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do Amazon MSK em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para obter mais informações, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como AWS CloudFormation. Para obter mais informações, consulte Elementos da política JSON do IAM: condição no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para obter mais informações, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter

mais informações, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para obter mais informações sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou 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": "*"
```

```
        }
    ]
}
```

Acessar um cluster do Amazon MSK

Neste exemplo, você vai permitir que um usuário do IAM na sua conta da Amazon Web Services acesse um dos seus cluster, `purchaseQueriesCluster`. Esta política permite que o usuário descreva o cluster, obtenha seus agentes de bootstrap, liste seus nós de agente e o atualize.

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"
      }
   ]
}
```

Como acessar clusters do Amazon MSK com base em tags

Você pode usar condições em sua política baseada em identidade para controlar o acesso aos recursos do Amazon MSK com base em tags. Este exemplo mostra como você pode criar uma política que permita que o usuário descreva o cluster, obtenha seus agentes de bootstrap, liste seus nós de agente, atualize-o e exclua-o. No entanto, a permissão será concedida somente se a tag de cluster `0wner` tiver o valor do nome desse usuário.

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}"
        }
      }
    }
  ]
}
```

É possível anexar essa política aos usuários do IAM na sua conta. Se um usuário chamado `richard-roe` tentar atualizar um cluster do MSK, o cluster deverá estar marcado como `0wner=richard-roe` ou `owner=richard-roe`. Caso contrário, ele terá o acesso negado. A chave da tag de condição `0wner` corresponde a `0wner` e a `owner` porque os nomes das chaves de condição não fazem distinção entre maiúsculas e minúsculas. Para obter mais informações, consulte IAM JSON Policy Elements: Condition (Elementos da política JSON do IAM: Condição) no Guia do usuário do IAM.

Perfis vinculados ao serviço para o Amazon MSK

O Amazon MSK usa funções vinculadas a serviços AWS Identity and Access Management (IAM). Um perfil vinculado a serviço é um tipo especial de perfil do IAM vinculado diretamente ao Amazon MSK. As funções vinculadas ao serviço são predefinidas pelo Amazon MSK e incluem todas as permissões que o serviço exige para chamar outros AWS serviços em seu nome.

Um perfil vinculado a serviço facilita a configuração do Amazon MSK porque você não precisa adicionar as permissões necessárias manualmente. O Amazon MSK define as permissões dos perfis vinculados a serviço. A menos que definido de outra forma, somente o Amazon MSK pode assumir seus perfis. As permissões definidas incluem a política de confiança e a política de permissões, que não pode ser anexada a nenhuma outra entidade do IAM.

Para obter informações sobre outros serviços compatíveis com perfis vinculados a serviço, consulte Serviços da Amazon Web Services compatíveis com o IAM e procure os serviços que exibem Sim na coluna Perfil vinculado a serviço. Escolha um Sim com um link para visualizar a documentação do perfil vinculado para esse serviço.

Tópicos

- Permissões de perfil vinculado a serviço para o Amazon MSK
- Criar um perfil vinculado ao serviço para o Amazon MSK
- Editar um perfil vinculado ao serviço para o Amazon MSK
- Regiões compatíveis com perfis vinculados a serviço do Amazon MSK

Permissões de perfil vinculado a serviço para o Amazon MSK

O Amazon MSK usa o perfil vinculado a serviço chamado AWSServiceRoleForKafka. O Amazon MSK usa esse perfil para acessar seus recursos e realizar operações como:

- `*NetworkInterface`: criar e gerenciar interfaces de rede na conta do cliente que tornem os agentes de cluster acessíveis aos clientes na VPC do cliente.
- `*VpcEndpoints`— gerencie endpoints de VPC na conta do cliente que tornam os agentes de cluster acessíveis aos clientes que usam a VPC do cliente. AWS PrivateLink O Amazon MSK usa permissões para `DescribeVpcEndpoints`, `ModifyVpcEndpoint` e `DeleteVpcEndpoints`.
- `secretsmanager`— gerencie as credenciais do cliente com AWS Secrets Manager.
- `GetCertificateAuthorityCertificate`: recuperar o certificado para sua autoridade de certificação privada.

Essa função vinculada ao serviço é anexada à seguinte política gerenciada: `KafkaServiceRolePolicy`. Para obter atualizações dessa política, consulte KafkaServiceRolePolicy.

O perfil vinculado ao serviço AWSServiceRoleForKafka confia nos seguintes serviços para aceitar o perfil:

- `kafka.amazonaws.com`

A política de permissões do perfil permite que o Amazon MSK execute as seguintes ações nos recursos.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "ec2:CreateNetworkInterface",
    "ec2:DescribeNetworkInterfaces",
    "ec2:CreateNetworkInterfacePermission",
    "ec2:AttachNetworkInterface",
    "ec2:DeleteNetworkInterface",
    "ec2:DetachNetworkInterface",
    "ec2:DescribeVpcEndpoints",
    "acm-pca:GetCertificateAuthorityCertificate",
    "secretsmanager:ListSecrets"
   ],
   "Resource": "*"
  },
  {
   "Effect": "Allow",
   "Action": [
    "ec2:ModifyVpcEndpoint"
   ],
   "Resource": "arn:*:ec2:*:*:subnet/*"
  },
  {
   "Effect": "Allow",
   "Action": [
    "ec2:DeleteVpcEndpoints",
    "ec2:ModifyVpcEndpoint"
   ],
   "Resource": "arn:*:ec2:*:*:vpc-endpoint/*",
```

```
    "Condition": {
     "StringEquals": {
      "ec2:ResourceTag/AWSMSKManaged": "true"
     },
     "StringLike": {
      "ec2:ResourceTag/ClusterArn": "*"
     }
    }
   },
   {
    "Effect": "Allow",
    "Action": [
     "secretsmanager:GetResourcePolicy",
     "secretsmanager:PutResourcePolicy",
     "secretsmanager:DeleteResourcePolicy",
     "secretsmanager:DescribeSecret"
    ],
    "Resource": "*",
    "Condition": {
     "ArnLike": {
      "secretsmanager:SecretId": "arn:*:secretsmanager:*:*:secret:AmazonMSK_*"
     }
    }
   }
  ]
}
```

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua um perfil vinculado a serviço. Para mais informações, consulte Permissões de perfil vinculado ao serviço no Guia do usuário do IAM.

Criar um perfil vinculado ao serviço para o Amazon MSK

Não é necessário criar uma função vinculada ao serviço manualmente. Quando você cria um cluster do Amazon MSK na AWS Management Console, na ou na AWS API AWS CLI, o Amazon MSK cria a função vinculada ao serviço para você.

Se excluir esse perfil vinculado ao serviço e precisar criá-lo novamente, será possível usar esse mesmo processo para recriar o perfil em sua conta. Quando você cria um cluster do Amazon MSK, o Amazon MSK cria um perfil vinculado a serviço para você novamente.

Editar um perfil vinculado ao serviço para o Amazon MSK

O Amazon MSK não permite que você edite o perfil vinculado a serviço do AWSServiceRoleForKafka. Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ele. No entanto, será possível editar a descrição do perfil usando o IAM. Para obter mais informações, consulte Editar uma função vinculada a serviço no Guia do usuário do IAM.

Regiões compatíveis com perfis vinculados a serviço do Amazon MSK

O Amazon MSK é compatível com perfis vinculados a serviço em todas as regiões nas quais o serviço esteja disponível. Para mais informações, consulte Regiões e endpoints da AWS.

AWS políticas gerenciadas para o Amazon MSK

Uma política AWS gerenciada é uma política autônoma criada e administrada por AWS. AWS as políticas gerenciadas são projetadas para fornecer permissões para muitos casos de uso comuns, para que você possa começar a atribuir permissões a usuários, grupos e funções.

Lembre-se de que as políticas AWS gerenciadas podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque estão disponíveis para uso de todos os AWS clientes. Recomendamos que você reduza ainda mais as permissões definindo as políticas gerenciadas pelo cliente que são específicas para seus casos de uso.

Você não pode alterar as permissões definidas nas políticas AWS gerenciadas. Se AWS atualizar as permissões definidas em uma política AWS gerenciada, a atualização afetará todas as identidades principais (usuários, grupos e funções) às quais a política está anexada. AWS é mais provável que atualize uma política AWS gerenciada quando uma nova AWS service (Serviço da AWS) é lançada ou novas operações de API são disponibilizadas para serviços existentes.

Para mais informações, consulte Políticas gerenciadas pela AWS no Manual do usuário do IAM.

AWS política gerenciada: Amazon MSKFull Access

Essa política concede permissões administrativas que permitem que a entidade principal tenha acesso total a todas as ações do Amazon MSK. As permissões nessa política são agrupadas da seguinte forma:

- As permissões do Amazon MSK permitem todas as ações do Amazon MSK.
- Permissões do **`Amazon EC2`**: nesta política são necessárias para validar os recursos passados em uma solicitação de API. Isso serve para garantir que o Amazon MSK seja capaz de usar

adequadamente os recursos com um cluster. O restante das EC2 permissões da Amazon nesta política permitem que o Amazon MSK crie os AWS recursos necessários para possibilitar a conexão com seus clusters.

- Permissões do **`AWS KMS`**: são usadas durante as chamadas de API para validar os recursos transmitidos em uma solicitação. Elas são necessárias para que o Amazon MSK consiga usar a chave transmitida com o cluster do Amazon MSK.
- Permissões do **`CloudWatch Logs, Amazon S3, and Amazon Data Firehose`**: são necessárias para que o Amazon MSK possa garantir que os destinos de entrega de logs sejam acessíveis e válidos para o uso de logs do agente.
- Permissões do **`IAM`**: são necessárias para que o Amazon MSK possa criar uma perfil vinculado ao serviço na conta e para permitir que você passe um perfil de execução de serviço para o 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 gerenciada: Amazon MSKRead OnlyAccess

Essa política concede permissões de acesso somente leitura que permitem que os usuários visualizem informações no Amazon MSK. As entidades principais com essa política anexada não podem fazer nenhuma atualização ou excluir recursos existentes, nem criar novos recursos do Amazon MSK. Por exemplo, entidades principais com essas permissões podem visualizar a lista de clusters e configurações associadas à conta, mas não podem alterar a configuração ou as definições de nenhum cluster. As permissões nessa política são agrupadas da seguinte forma:

- Permissões do **`Amazon MSK`**: permitem que você liste os recursos do Amazon MSK, descreva-os e obtenha informações sobre eles.
- **`Amazon EC2`**permissões — são usadas para descrever a Amazon VPC, sub-redes, grupos de segurança e ENIs que estão associados a um cluster.
- Permissão do **`AWS KMS`**: é usada para descrever a chave associada ao cluster.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "kafka:Describe*",
                "kafka:List*",
```

```
                "kafka:Get*",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "kms:DescribeKey"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

AWS política gerenciada: KafkaServiceRolePolicy

Você não pode se vincular KafkaServiceRolePolicy às suas entidades do IAM. Essa política é anexada a um perfil vinculado a serviço que permite que o Amazon MSK realize ações como gerenciar endpoints da VPC (conectores) em clusters do MSK, gerenciar interfaces de rede e gerenciar credenciais de cluster com o AWS Secrets Manager. Para obter mais informações, consulte the section called “Perfis vinculados a serviço”.

AWS política gerenciada: AWSMSKReplicator ExecutionRole

A política `AWSMSKReplicatorExecutionRole` concede permissões ao Replicador do Amazon MSK para replicar dados entre clusters do MSK. As permissões nessa política são agrupadas da seguinte forma:

- **`cluster`**: concede ao Replicador do Amazon MSK permissões para se conectar ao cluster usando a autenticação do IAM. Também concede permissões para descrever e alterar o cluster.
- **`topic`**: concede ao Replicador do Amazon MSK permissões para descrever, criar e alterar um tópico e alterar a configuração dinâmica dele.
- **`consumer group`**: concede ao Replicador do Amazon MSK permissões para descrever e alterar grupos de consumidores, ler e gravar datas de um cluster do MSK e excluir tópicos internos criados pelo 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/*/*"
    ]
   }
  ]
}
```

Atualizações do Amazon MSK para políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas do Amazon MSK desde que esse serviço começou a rastrear essas alterações.

Alteração	Descrição	Data
WriteDataIdempotently permissão adicionada a AWSMSKReplicator Execution Role — Atualização de uma política existente	O Amazon MSK adicionou WriteDataIdempotently permissão à AWSMSKRep licator ExecutionRole política para oferecer suporte à replicação de dados entre clusters MSK.	12 de março de 2024
AWSMSKReplicatorEx ecutionRole – Nova política	O Amazon MSK adicionou uma AWSMSKReplicator ExecutionRole política para dar suporte ao Amazon MSK Replicator.	4 de dezembro de 2023
Amazon MSKFull Access — Atualização de uma política existente	O Amazon MSK adicionou permissões para compatibi lidade com o replicador do Amazon MSK.	28 de setembro de 2023

Alteração	Descrição	Data
KafkaServiceRolePolicy: atualização para uma política existente	O Amazon MSK adicionou permissões para compatibilidade com conectividade privada multi-VPC.	8 de março de 2023
Amazon MSKFull Access — Atualização de uma política existente	O Amazon MSK adicionou novas EC2 permissões da Amazon para possibilitar a conexão a um cluster.	30 de novembro de 2021
Amazon MSKFull Access — Atualização de uma política existente	O Amazon MSK adicionou uma nova permissão para permitir a descrição das tabelas de EC2 rotas da Amazon.	19 de novembro de 2021
O Amazon MSK passou a monitorar alterações	A Amazon MSK começou a monitorar as mudanças em suas políticas AWS gerenciadas.	19 de novembro de 2021

Solução de problemas de identidade e acesso do Amazon MKS

Use as informações a seguir para ajudar a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com o Amazon MSK e o IAM.

Tópicos

- Não tenho autorização para executar uma ação no Amazon MSK

Não tenho autorização para executar uma ação no Amazon MSK

Se isso AWS Management Console indicar que você não está autorizado a realizar uma ação, entre em contato com o administrador para obter ajuda. Caso seu administrador seja a pessoa que forneceu suas credenciais de início de sessão.

O exemplo de erro a seguir ocorre quando o usuário do IAM `mateojackson` tenta usar o console para excluir um cluster, mas não tem permissões `kafka:DeleteCluster`.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 kafka:DeleteCluster on resource: purchaseQueriesCluster
```

Neste caso, Mateo pede ao administrador para atualizar suas políticas para permitir a ele o acesso ao recurso `purchaseQueriesCluster` usando a ação `kafka:DeleteCluster`.

Autenticação e autorização para o Apache Kafka APIs

É possível usar o IAM para autenticar clientes e permitir ou proibir ações do Apache Kafka. Como alternativa, você pode usar o TLS ou SASL/SCRAM para autenticar clientes e o Apache Kafka ACLs para permitir ou negar ações.

Para obter informações sobre como controlar quem pode realizar operações do Amazon MSK em seu cluster, consulte the section called “Autenticação e autorização para Amazon MSK APIs”.

Tópicos

- Controle de acesso do IAM
- Autenticação mútua de cliente TLS para o Amazon MSK
- Autenticação de credenciais de login com Secrets Manager AWS
- Apache Kafka ACLs

Controle de acesso do IAM

O controle de acesso do IAM para o Amazon MSK permite que você gerencie a autenticação e a autorização para seu cluster do MSK. Isso elimina a necessidade de usar um mecanismo para autenticação e outro para autorização. Por exemplo, quando um cliente tenta gravar em seu cluster, o Amazon MSK usa o IAM para verificar se esse cliente é uma identidade autenticada e também se ele está autorizado a produzir para seu cluster.

O controle de acesso do IAM funciona para clientes Java e não Java, incluindo clientes Kafka escritos em Python, Go e .NET. JavaScript O controle de acesso do IAM para clientes não Java está disponível para clusters MSK com a versão 2.7.1 ou superior do Kafka.

Para viabilizar o controle de acesso do IAM, o Amazon MSK faz pequenas modificações no código-fonte do Apache Kafka. Essas modificações não causarão uma diferença perceptível na sua

experiência com o Apache Kafka. O Amazon MSK registra eventos de acesso para que você possa auditá-los.

Você pode invocar o Apache Kafka ACL APIs para um cluster MSK que usa o controle de acesso do IAM. No entanto, o Apache Kafka não ACLs tem efeito na autorização de identidades do IAM. Você deve usar políticas do IAM para controlar o acesso às identidades do IAM.

⚠ Considerações importantes

Ao usar o controle de acesso do IAM com seu cluster MSK, lembre-se das seguintes considerações importantes:

- O controle de acesso do IAM não se aplica aos ZooKeeper nós do Apache. Para obter informações sobre como você pode controlar o acesso a esses nós, consulte Controle o acesso aos ZooKeeper nós do Apache em seu cluster Amazon MSK.
- A configuração `allow.everyone.if.no.acl.found` do Apache Kafka não tem efeito se seu cluster usar o controle de acesso do IAM.
- Você pode invocar o Apache Kafka ACL APIs para um cluster MSK que usa o controle de acesso do IAM. No entanto, o Apache Kafka não ACLs tem efeito na autorização de identidades do IAM. Você deve usar políticas do IAM para controlar o acesso às identidades do IAM.

Funcionamento do controle de acesso do IAM para o Amazon MSK

Para usar o controle de acesso do IAM para o Amazon MSK, execute as etapas a seguir, descritas em mais detalhes nestes tópicos:

- Criar um cluster do Amazon MSK que use o controle de acesso do IAM
- Configurar clientes para controle de acesso do IAM
- Criar políticas de autorização para o perfil do IAM
- Obter os agente de bootstrap para controle de acesso do IAM

Criar um cluster do Amazon MSK que use o controle de acesso do IAM

Esta seção explica como você pode usar a AWS Management Console API ou a AWS CLI para criar um cluster Amazon MSK que usa o controle de acesso do IAM. Para obter informações sobre como

ativar o controle de acesso do IAM para um cluster existente, consulte Atualizar as configurações de segurança de um cluster do Amazon MSK.

Use o AWS Management Console para criar um cluster que usa o controle de acesso do IAM

1. Abra o console do Amazon MSK em https://console.aws.amazon.com/msk/.
2. Selecione Criar cluster.
3. Escolha Criar cluster com configurações personalizadas.
4. Na seção Autenticação, escolha Controle de acesso do IAM.
5. Preencha o restante do fluxo de trabalho para criar um cluster.

Use a API ou a AWS CLI para criar um cluster que usa o controle de acesso do IAM

- Para criar um cluster com o controle de acesso IAM ativado, use a CreateClusterAPI ou o comando da CLI create-cluster e passe o seguinte JSON para o parâmetro:. `ClientAuthentication` `"ClientAuthentication": { "Sasl": { "Iam": { "Enabled": true } }`

Configurar clientes para controle de acesso do IAM

Para permitir que os clientes se comuniquem com um cluster do MSK que use o controle de acesso do IAM, você pode usar um dos seguintes mecanismos:

- Configuração de cliente não Java usando mecanismo SASL_OAUTHBEARER
- Configuração do cliente Java usando SASL_OAUTHBEARER mecanismo ou AWS_MSK_IAM mecanismo

Use o SASL_OAUTHBEARER mecanismo para configurar o IAM

1. Edite seu arquivo de configuração client.properties usando o seguinte exemplo de cliente Python Kafka. As alterações das configurações são semelhantes em outros 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 Região da 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. Faça o download da biblioteca auxiliar para o idioma de configuração escolhido e siga as instruções na seção Introdução da página inicial dessa biblioteca de idiomas.

   - JavaScript: https://github.com/aws/aws-msk-iam-sasl-signer-js #getting -iniciado
   - Python: https://github.com/aws/aws-msk-iam-sasl-signer-python #get -started
   - Go: https://github.com/aws/aws-msk-iam-sasl-signer-go #getting -started
   - .NET: https://github.com/aws/aws-msk-iam-sasl-signer-net #getting -iniciado
   - JAVA: o SASL_OAUTHBEARER suporte para Java está disponível por meio do arquivo `aws-msk-iam-auth`jar

Use o AWS_MSK_IAM mecanismo personalizado do MSK para configurar o IAM

1. Adicione o seguinte ao arquivo `client.properties`. *<PATH_TO_TRUST_STORE_FILE>*Substitua pelo caminho totalmente qualificado para o arquivo de armazenamento confiável no cliente.

   > **Note**
   >
   > Se você não quiser usar um certificado específico, poderá remover `ssl.truststore.location=`*<PATH_TO_TRUST_STORE_FILE>* do seu arquivo `client.properties`. Se você não especificar um valor para `ssl.truststore.location`, o processo Java usará o certificado padrão.

   ```
   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 um perfil nomeado que você criou para AWS credenciais, inclua `awsProfileName="`*your profile name*`";` no arquivo de configuração do cliente. Para obter informações sobre perfis nomeados, consulte Perfis nomeados na AWS CLI documentação.

2. Baixe o arquivo aws-msk-iam-authJAR estável mais recente e coloque-o no caminho da classe. Se você usa o Maven, adicione a seguinte dependência, ajustando o número da versão conforme necessário:

   ```
   <dependency>
       <groupId>software.amazon.msk</groupId>
       <artifactId>aws-msk-iam-auth</artifactId>
       <version>1.0.0</version>
   </dependency>
   ```

O plug-in do cliente do Amazon MSK é de código aberto sob a licença do Apache 2.0.

Criar políticas de autorização para o perfil do IAM

Anexe uma política de autorização ao perfil do IAM correspondente ao cliente. Em uma política de autorização, você especifica quais ações permitir ou proibir para o perfil. Se seu cliente estiver em uma EC2 instância da Amazon, associe a política de autorização à função do IAM dessa EC2 instância da Amazon. Como alternativa, você pode configurar seu cliente para usar um perfil nomeado e, em seguida, associar a política de autorização ao perfil desse perfil nomeado. Configurar clientes para controle de acesso do IAM descreve como configurar um cliente para usar um perfil nomeado.

Para obter informações sobre como criar uma política do IAM, consulte Criar políticas do IAM.

Veja a seguir um exemplo de política de autorização para um cluster chamado MyTestCluster. Para entender a semântica dos elementos `Action` e `Resource`, consulte Semântica das ações e recursos da política de autorização do IAM.

> **Important**
>
> As alterações que você faz em uma política do IAM são refletidas no IAM APIs e AWS CLI imediatamente. No entanto, a implementação da alteração da política pode levar um tempo considerável. Na maioria dos casos, as mudanças na política entram em vigor em menos de um minuto. Às vezes, as condições da rede podem aumentar o atraso.

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 saber como criar uma política com elementos de ação que correspondam aos casos de uso comuns do Apache Kafka, como produzir e consumir dados, consulte Casos de uso comuns para a política de autorização de clientes.

Para as versões 2.8.0 e superiores do Kafka, a WriteDataIdempotentlypermissão está obsoleta (KIP-679). `enable.idempotence = true` é usado por padrão. Portanto, para as versões 2.8.0 e superiores do Kafka, o IAM não oferece a mesma funcionalidade do Kafka. ACLs Não é possível `WriteDataIdempotently` acessar um tópico fornecendo `WriteData` acesso apenas a esse tópico. Isso não afeta o caso quando `WriteData` é fornecido para TODOS os tópicos. Nesse caso, `WriteDataIdempotently` é permitido. Isso se deve às diferenças na implementação da lógica do IAM e na forma como os Kafka ACLs são implementados. Além disso, escrever em um tópico de forma idempotente também requer acesso a. `transactional-ids`

Para contornar isso, recomendamos o uso de uma política semelhante à política a seguir.

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/*"
            ]
        }
    ]
}
```

Nesse caso, `WriteData` permite gravações em `TestTopic`, enquanto `WriteDataIdempotently` permite gravações idempotentes no cluster. Essa política também adiciona acesso aos `transactional-id` recursos que serão necessários.

Como `WriteDataIdempotently` é uma permissão no nível do cluster, você não pode usá-la no nível do tópico. Se `WriteDataIdempotently` estiver restrita ao nível do tópico, essa política não funcionará.

Obter os agente de bootstrap para controle de acesso do IAM

Consulte Obter os agentes de bootstrap para um cluster do Amazon MSK.

Semântica das ações e recursos da política de autorização do IAM

Atualmente, o controle de acesso do IAM para o Amazon MSK não oferece suporte a ações internas de cluster para o Kafka. Isso inclui a WriteTxnMarkers API, que o Kafka usa para encerrar transações. Para encerrar transações, recomendamos que você use a autenticação SCRAM ou mTLS com a autenticação apropriada, ACLs em vez da autenticação IAM.

Esta seção explica a semântica dos elementos de ação e recurso que você pode usar em uma política de autorização do IAM. Para visualizar um exemplo de política, consulte Criar políticas de autorização para o perfil do IAM.

Ações da política de autorização

A tabela a seguir lista as ações que você pode incluir em uma política de autorização ao usar o controle de acesso do IAM para o Amazon MSK. Ao incluir uma ação da coluna Ação da tabela em sua política de autorização, você também deve incluir as ações correspondentes da coluna Ações obrigatórias.

Ação	Descrição	Ações necessárias	Recursos necessários	Aplicável a clusters com a tecnologia sem servidor
`kafka-cluster:Connect`	Concede permissão para se conectar e se autenticar no cluster.	Nenhum	cluster	Sim
`kafka-cluster:Desc`	Concede permissão para descrever	`kafka-cluster:Connect`	cluster	Sim

Ação	Descrição	Ações necessárias	Recursos necessários	Aplicável a clusters com a tecnologia sem servidor
`ribeCluster`	vários aspectos do cluster, equivalente à ACL DESCRIBE CLUSTER do Apache Kafka.			

`kafka-cluster:AlterCluster`	Concede permissão para alterar vários aspectos do cluster, equivalente à ACL ALTER CLUSTER do Apache Kafka.	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeCluster`
			cluster	Não

`kafka-cluster:DescribeClusterDynamicConfiguration`	Concede permissão para descrever a configuração dinâmica de um cluster, equivalente à ACL DESCRIBE_CONFIGS CLUSTER do Apache Kafka.	`kafka-cluster:Connect`	cluster	Não

Ação	Descrição	Ações necessárias	Recursos necessários	Aplicável a clusters com a tecnologia sem servidor

`kafka-cluster:AlterClusterDynamicConfiguration`	Concede permissão para alterar a configuração dinâmica de um cluster, equivalente à ACL ALTER_CONFIGS CLUSTER do Apache Kafka.	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeClusterDynamicConfiguration`
			cluster	Não

`kafka-cluster:WriteDataIdempotently`	Concede permissão para gravar dados em um cluster de modo idempotente, equivalente à ACL IDEMPOTENT_WRITE CLUSTER do Apache Kafka.	`kafka-cluster:Connect`
		
		`kafka-cluster:WriteData`
			cluster	Sim

Ação	Descrição	Ações necessárias	Recursos necessários	Aplicável a clusters com a tecnologia sem servidor
`kafka-cluster:CreateTopic`	Concede permissão para criar tópicos em um cluster, equivalente à CREATE ACL do Apache Kafka. CLUSTER/TOPIC	`kafka-cluster:Connect`	tópico	Sim
`kafka-cluster:DescribeTopic`	Concede permissão para descrever os tópicos de um cluster, equivalente à ACL DESCRIBE TOPIC do Apache Kafka.	`kafka-cluster:Connect`	tópico	Sim
`kafka-cluster:AlterTopic`	Concede permissão para alterar os tópicos de um cluster, equivalente à ACL ALTER TOPIC do Apache Kafka.	`kafka-cluster:Connect` `kafka-cluster:DescribeTopic`	tópico	Sim

Ação	Descrição	Ações necessárias	Recursos necessários	Aplicável a clusters com a tecnologia sem servidor
`kafka-cluster:DeleteTopic`	Concede permissão para excluir tópicos de um cluster, equivalente à ACL DELETE TOPIC do Apache Kafka.	`kafka-cluster:Connect` `kafka-cluster:DescribeTopic`	tópico	Sim
`kafka-cluster:DescribeTopicDynamicConfiguration`	Concede permissão para descrever a configuração dinâmica dos tópicos de um cluster, equivalente à ACL DESCRIBE_CONFIGS TOPIC do Apache Kafka.	`kafka-cluster:Connect`	tópico	Sim

Ação	Descrição	Ações necessárias	Recursos necessários	Aplicável a clusters com a tecnologia sem servidor
`kafka-cluster:AlterTopicDynamicConfiguration`	Concede permissão para alterar a configuração dinâmica dos tópicos de um cluster, equivalente à ACL ALTER_CONFIGS TOPIC do Apache Kafka.	`kafka-cluster:Connect` `kafka-cluster:DescribeTopicDynamicConfiguration`	tópico	Sim

`kafka-cluster:ReadData`	Concede permissão para ler dados dos tópicos de um cluster, equivalente à ACL READ TOPIC do Apache Kafka.	`kafka-cluster:Connect`
		`kafka-cluster:DescribeTopic`
		`kafka-cluster:AlterGroup`
			tópico	Sim

Ação	Descrição	Ações necessárias	Recursos necessários	Aplicável a clusters com a tecnologia sem servidor

`kafka-cluster:WriteData`	Concede permissão para gravar dados em tópicos de um cluster, equivalente a WRITE TOPIC ACL do Apache Kafka.	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeTopic`
			tópico	Sim

`kafka-cluster:DescribeGroup`	Concede permissão para descrever os grupos de um cluster, equivalente à ACL DESCRIBE GROUP do Apache Kafka.	`kafka-cluster:Connect`	grupo	Sim

`kafka-cluster:AlterGroup`	Concede permissão para entrar em grupos de um cluster, equivalente à ACL READ GROUP do Apache Kafka.	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeGroup`
			grupo	Sim

Ação	Descrição	Ações necessárias	Recursos necessários	Aplicável a clusters com a tecnologia sem servidor
`kafka-cluster:DeleteGroup`	Concede permissão para excluir grupos de um cluster, equivalente à ACL DELETE GROUP do Apache Kafka.	`kafka-cluster:Connect` `kafka-cluster:DescribeGroup`	grupo	Sim
`kafka-cluster:DescribeTransactionalId`	Concede permissão para descrever transações IDs em um cluster, equivalente à ACL DESCRIBE TRANSACTIONAL_ID do Apache Kafka.	`kafka-cluster:Connect`	transactional-id	Sim

`kafka-cluster:AlterTransactionalId`	Concede permissão para alterar a transação IDs em um cluster, equivalente à ACL WRITE TRANSACTIONAL_ID do Apache Kafka.	`kafka-cluster:Connect`
		`kafka-cluster:DescribeTransactionalId`
		`kafka-cluster:WriteData`
			transactional-id	Sim

Você pode usar o curinga asterisco (*) quantas vezes quiser em uma ação após o sinal de dois pontos. Veja os exemplos a seguir.

- `kafka-cluster:*Topic` corresponde a `kafka-cluster:CreateTopic`, `kafka-cluster:DescribeTopic`, `kafka-cluster:AlterTopic` e `kafka-cluster:DeleteTopic`. Isso não inclui `kafka-cluster:DescribeTopicDynamicConfiguration` ou `kafka-cluster:AlterTopicDynamicConfiguration`.
- `kafka-cluster:*` corresponde a todas as permissões.

Recursos da política de autorização

A tabela a seguir mostra os quatro tipos de recurso que você pode usar em uma política de autorização ao usar o controle de acesso do IAM para o Amazon MSK. Você pode obter o Amazon Resource Name (ARN) do cluster no AWS Management Console ou usando a DescribeClusterAPI ou o comando AWS CLI describe-cluster. Em seguida, você pode usar o ARN do cluster para criar ID de tópico, grupo e transação. ARNs Para especificar um recurso em uma política de autorização, use o ARN desse recurso.

Recurso	Formato ARN
Cluster	arn:aws:kafka: ::cluster//*regionaccount-id cluster-name cluster-uuid*
Tópico	ar:aws:kafka: ::topic///*regionaccount-id cluster-name cluster-uuid topic-name*
Grupo	arn:aws:kafka: ::group///*regionaccount-id cluster-name cluster-uuid group-name*
ID transacional	arn:aws:kafka: ::transactional-id///*regionaccount-id cluster-name cluster-uuid transactional-id*

Você pode usar o curinga asterisco (*) quantas vezes quiser em qualquer lugar na parte do ARN que vem depois de `:cluster/`, `:topic/`, `:group/` e `:transactional-id/`. Veja a seguir alguns exemplos de como usar o curinga asterisco (*) para se referir a vários recursos:

- `arn:aws:kafka:us-east-1:0123456789012:topic/MyTestCluster/*`: todos os tópicos em qualquer cluster nomeado MyTestCluster, independentemente do UUID do cluster.
- `arn:aws:kafka:us-east-1:0123456789012:topic/MyTestCluster/abcd1234-0123-abcd-5678-1234abcd-1/*_test`: todos os tópicos cujo nome termina com "_test" no cluster cujo nome é MyTestCluster e cujo UUID é abcd1234-0123-abcd-5678-1234abcd-1.
- `arn:aws:kafka:us-east-1:0123456789012:transactional-id/MyTestCluster/*/5555abcd-1111-abcd-1234-abcd1234-1`: todas as transações cuja ID transacional é 5555abcd-1111-abcd-1234-abcd1234-1, em todas as encarnações de um cluster nomeado em sua conta. MyTestCluster Isso significa que, se você criar um cluster chamado MyTestCluster, excluí-lo e criar outro cluster com o mesmo nome, poderá usar esse ARN de recurso para representar a mesma ID de transação nos dois clusters. No entanto, o cluster excluído não estará acessível.

Casos de uso comuns para a política de autorização de clientes

A primeira coluna na tabela a seguir mostra alguns casos de uso comuns. Para autorizar um cliente a executar um determinado caso de uso, inclua as ações necessárias para esse caso de uso na política de autorização do cliente e defina `Effect` como `Allow`.

Para obter informações sobre todas as ações que fazem parte do controle de acesso do IAM para o Amazon MSK, consulte Semântica das ações e recursos da política de autorização do IAM.

> **Note**
>
> As ações são negadas por padrão. Você deve permitir explicitamente todas as ações que deseja autorizar o cliente a executar.

Caso de uso	Ações necessárias
Administrador	`kafka-cluster:*`
Criar um tópico	`kafka-cluster:Connect` `kafka-cluster:CreateTopic`
Produzir dados	`kafka-cluster:Connect` `kafka-cluster:DescribeTopic`

Caso de uso	Ações necessárias
	`kafka-cluster:WriteData`

Consumir dados	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:DescribeGroup`
	`kafka-cluster:AlterGroup`
	`kafka-cluster:ReadData`

Produzir dados de modo idempotente	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:WriteData`
	`kafka-cluster:WriteDataIdempotently`

Produzir dados de modo transacional	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:WriteData`
	`kafka-cluster:DescribeTransactionalId`
	`kafka-cluster:AlterTransactionalId`

Descrever a configuração de um cluster	`kafka-cluster:Connect` `kafka-cluster:DescribeClusterDynamicConfiguration`

Caso de uso	Ações necessárias

Atualizar a configuração de um cluster	`kafka-cluster:Connect`
	
	`kafka-cluster:DescribeClusterDynamicConfiguration`
	
	`kafka-cluster:AlterClusterDynamicConfiguration`

Descrever a configuração de um tópico	`kafka-cluster:Connect`
	
	`kafka-cluster:DescribeTopicDynamicConfiguration`

Atualizar a configuração de um tópico	`kafka-cluster:Connect`
	
	`kafka-cluster:DescribeTopicDynamicConfiguration`
	
	`kafka-cluster:AlterTopicDynamicConfiguration`

Alterar um tópico	`kafka-cluster:Connect`
	
	`kafka-cluster:DescribeTopic`
	
	`kafka-cluster:AlterTopic`

Autenticação mútua de cliente TLS para o Amazon MSK

Você pode habilitar a autenticação do cliente com TLS para conexões das aplicações aos agentes do Amazon MSK. Para usar a autenticação do cliente, é necessário ter um CA privada da AWS. Eles CA privada da AWS podem estar no Conta da AWS mesmo cluster ou em uma conta diferente. Para obter informações sobre CA privada da AWS s, consulte Criando e gerenciando um CA privada da AWS.

> **Note**
>
> No momento, a autenticação por TLS não está disponível nas regiões Pequim e Ningxia.

O Amazon MSK não oferece suporte a listas de revogação de certificados (). CRLs Para controlar o acesso aos tópicos do cluster ou bloquear certificados comprometidos, use o Apache Kafka ACLs e grupos de segurança. AWS Para obter informações sobre como usar o Apache Kafka ACLs, consulte. the section called “Apache Kafka ACLs”

Este tópico contém as seguintes seções:

- Criar um cluster do Amazon MSK compatível com a autenticação de cliente
- Configurar um cliente para usar a autenticação
- Produzir e consumir mensagens usando autenticação

Criar um cluster do Amazon MSK compatível com a autenticação de cliente

Este procedimento mostra como habilitar a autenticação do cliente usando um CA privada da AWS.

> **Note**
>
> É altamente recomendável usar o independente CA privada da AWS para cada cluster MSK ao usar o TLS mútuo para controlar o acesso. Isso garantirá que os certificados TLS assinados por sejam autenticados PCAs apenas com um único cluster MSK.

1. Crie um arquivo denominado `clientauthinfo.json` com o seguinte conteúdo: *`Private-CA-ARN`*Substitua pelo ARN do seu PCA.

   ```
   {
      "Tls": {
          "CertificateAuthorityArnList": ["Private-CA-ARN"]
       }
   }
   ```

2. Crie um arquivo chamado `brokernodegroupinfo.json`, conforme descrito em the section called “Crie um cluster Amazon MSK provisionado usando o AWS CLI”.

3. A autenticação de cliente exige que você também ative a criptografia em trânsito entre clientes e agentes. Crie um arquivo denominado `encryptioninfo.json` com o seguinte conteúdo: *KMS-Key-ARN*Substitua pelo ARN da sua chave KMS. É possível definir `ClientBroker` como `TLS` ou `TLS_PLAINTEXT`.

```
{
   "EncryptionAtRest": {
       "DataVolumeKMSKeyId": "KMS-Key-ARN"
    },
   "EncryptionInTransit": {
        "InCluster": true,
        "ClientBroker": "TLS"
    }
}
```

   Para obter mais informações sobre criptografia, consulte the section called “Criptografia do Amazon MSK”.

4. Em uma máquina em que você tenha o AWS CLI instalado, execute o comando a seguir para criar um cluster com autenticação e criptografia em trânsito ativadas. Salve o ARN do cluster fornecido na resposta.

```
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
```

Configurar um cliente para usar a autenticação

Esse processo descreve como configurar uma EC2 instância da Amazon para usar como cliente para usar a autenticação.

Este processo descreve como produzir e consumir mensagens usando a autenticação criando uma máquina cliente, criando um tópico e definindo as configurações de segurança necessárias.

1. Crie uma EC2 instância da Amazon para usar como máquina cliente. Para simplificar, crie essa instância na mesma VPC usada para o cluster. Consulte the section called “Criar uma máquina cliente” para obter um exemplo de como criar uma máquina de cliente.
2. Criar um tópico. Para obter um exemplo, consulte as instruções em the section called “Criar um tópico”.

3. Em uma máquina em que você tem o AWS CLI instalado, execute o comando a seguir para obter os corretores de bootstrap do cluster. *Cluster-ARN*Substitua pelo ARN do seu cluster.

```
aws kafka get-bootstrap-brokers --cluster-arn Cluster-ARN
```

   Salve a string associada ao `BootstrapBrokerStringTls` na resposta.

4. Na máquina de cliente, execute o comando a seguir para usar o armazenamento de confiança da JVM para criar o armazenamento de confiança do cliente. Se o caminho da JVM for diferente, ajuste o comando de acordo.

```
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. Na máquina de cliente, execute o comando a seguir para criar uma chave privada para o cliente. Substitua *Distinguished-NameExample-Alias*,*Your-Store-Pass*,, e *Your-Key-Pass* por cordas de sua escolha.

```
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. Na máquina de cliente, execute o comando a seguir para criar uma solicitação de certificado com a chave privada criada na etapa 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 o arquivo `client-cert-sign-request` e verifique se ele começa com `-----BEGIN CERTIFICATE REQUEST-----` e termina com `-----END CERTIFICATE REQUEST-----`. Se ele começar com `-----BEGIN NEW CERTIFICATE REQUEST-----`, exclua a palavra `NEW` (e o espaço único que vem após) do começo e do final do arquivo.

8. Em uma máquina em que você tenha o AWS CLI instalado, execute o comando a seguir para assinar sua solicitação de certificado. *Private-CA-ARN*Substitua pelo ARN do seu PCA. Será possível alterar o valor de validade se quiser. Aqui usamos 300 como exemplo.

```
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"
```

Salve o ARN do certificado fornecido na resposta.

> **Note**
>
> Para recuperar seu certificado de cliente, use o comando `acm-pca get-certificate` e especifique o ARN do certificado. Para obter mais informações, consulte get-certificate na Referência de comandos da AWS CLI .

9. Execute o comando a seguir para obter o certificado CA privada da AWS assinado para você. *`Certificate-ARN`*Substitua pelo ARN obtido da resposta ao comando anterior.

```
aws acm-pca get-certificate --certificate-authority-arn Private-CA-ARN --
certificate-arn Certificate-ARN
```

10. Do resultado JSON obtido com a execução do comando anterior, copie as strings associadas a `Certificate` e `CertificateChain`. Cole essas duas sequências em um novo arquivo chamado signed-certificate-from-acm. Cole a string associada a `Certificate` primeiro, seguida pela string associada a `CertificateChain`. Substitua os caracteres `\n` por novas linhas. Veja a seguir a estrutura do arquivo depois que você colar o certificado e a cadeia de certificados nele.

```
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
```

11. Execute o comando a seguir na máquina cliente para adicionar esse certificado ao repositório de chaves para poder apresentá-lo ao falar com os agentes do 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. Crie um arquivo denominado `client.properties` com o seguinte conteúdo: Ajuste os locais do armazenamento de confiança e do repositório de chaves usando os caminhos onde salvou

`kafka.client.truststore.jks`. Substitua os espaços reservados por sua versão do cliente 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
```

Produzir e consumir mensagens usando autenticação

Este processo descreve como produzir e consumir mensagens usando a autenticação.

1. Execute o comando a seguir para criar um tópico. O arquivo chamado `client.properties` é o que você criou no procedimento 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. Execute o comando a seguir para iniciar um produtor de console. O arquivo chamado `client.properties` é o que você criou no procedimento anterior.

```
<path-to-your-kafka-installation>/bin/kafka-console-producer.sh --bootstrap-
server BootstrapBroker-String --topic ExampleTopic --producer.config
 client.properties
```

3. Em uma nova janela de comando na máquina de cliente, execute o comando a seguir para iniciar um consumidor de console.

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBroker-String --topic ExampleTopic --consumer.config
 client.properties
```

4. Digite mensagens na janela do produtor e observe-as aparecerem na janela do consumidor.

Autenticação de credenciais de login com Secrets Manager AWS

Você pode controlar o acesso aos seus clusters do Amazon MSK usando credenciais de login que são armazenadas e protegidas usando o Secrets Manager. AWS Armazenar as credenciais de usuário no Secrets Manager reduz a sobrecarga da autenticação do cluster, como auditoria, atualização e rodízio de credenciais. O Secrets Manager também permite que você compartilhe credenciais de usuário entre clusters.

Depois de associar um segredo a um cluster MSK, o MSK sincroniza os dados da credencial periodicamente.

Este tópico contém as seguintes seções:

- Como a autenticação de credenciais de acesso funciona
- Configurar a SASL/SCRAM autenticação para um cluster Amazon MSK
- Trabalhar com usuários
- Limitações ao usar segredos do SCRAM

Como a autenticação de credenciais de acesso funciona

A autenticação de credenciais de login para o Amazon MSK usa a autenticação SASL/SCRAM (Simple Authentication and Security Layer/Salted Challenge Response Mechanism). Para configurar a autenticação de credenciais de acesso para um cluster, você cria um recurso secreto no AWS Secrets Manager e associa as credenciais de acesso a esse segredo.

O SASL/SCRAM está definido no RFC 5802. O SCRAM usa algoritmos de hash protegidos e não transmite credenciais em texto simples entre o cliente e o servidor.

> **Note**
>
> Quando você configura a SASL/SCRAM autenticação para seu cluster, o Amazon MSK ativa a criptografia TLS para todo o tráfego entre clientes e agentes.

Configurar a SASL/SCRAM autenticação para um cluster Amazon MSK

Para configurar um segredo no AWS Secrets Manager, siga o tutorial Criando e recuperando um segredo no Guia do usuário do AWS Secrets Manager.

Observe os seguintes requisitos ao criar um segredo para um cluster do Amazon MSK:

- Escolha Outros tipos de segredos (p. ex., chave de API) para o tipo de segredo.
- O nome do segredo deve começar com o prefixo AmazonMSK_.
- Você deve usar uma AWS KMS chave personalizada existente ou criar uma nova AWS KMS chave personalizada para seu segredo. O Secrets Manager usa a AWS KMS chave padrão para um segredo por padrão.

> **Important**
>
> Um segredo criado com a AWS KMS chave padrão não pode ser usado com um cluster Amazon MSK.

- Seus dados de credencial de acesso devem estar no formato a seguir para que seja possível inserir pares de valor/chave usando a opção Texto simples.

```
{
  "username": "alice",
  "password": "alice-secret"
}
```

- Registre o valor do ARN (nome do recurso da Amazon) do seu segredo.
-

> **Important**
>
> Você não pode associar um segredo do Secrets Manager a um cluster que exceda os limites descritos em the section called “ Dimensione seu cluster adequadamente: número de partições por agente padrão”.

- Se você usar o AWS CLI para criar o segredo, especifique um ID de chave ou ARN para o `kms-key-id` parâmetro. Não especifique um alias.
- Para associar o segredo ao seu cluster, use o console Amazon MSK ou a BatchAssociateScramSecretoperação.

> **Important**
>
> Quando você associa um segredo a um cluster, o Amazon MSK anexa uma política de recursos ao segredo, permitindo que seu cluster acesse e leia os valores secretos que você definiu. Você não deve modificar essa política de recursos. Isso pode impedir que seu cluster acesse seu segredo. Se você fizer alguma alteração na política de recursos de segredos e/ou na chave KMS usada para criptografia secreta, certifique-se de

associar novamente os segredos ao seu cluster MSK. Isso garantirá que seu cluster possa continuar acessando seu segredo.

O exemplo de entrada JSON a seguir para a operação `BatchAssociateScramSecret` associa um segredo a um cluster:

```
{
  "clusterArn" : "arn:aws:kafka:us-west-2:0123456789019:cluster/SalesCluster/
abcd1234-abcd-cafe-abab-9876543210ab-4",
  "secretArnList": [
    "arn:aws:secretsmanager:us-west-2:0123456789019:secret:AmazonMSK_MyClusterSecret"
  ]
}
```

Como estabelecer conexão com o seu cluster usando credenciais de acesso

Após criar um segredo e associá-lo ao cluster, você poderá conectar o cliente ao cluster. O procedimento a seguir demonstra como conectar um cliente a um cluster que usa SASL/SCRAM autenticação. Também mostra como produzir e consumir a partir de um tópico de exemplo.

Tópicos

- Conectando um cliente ao cluster usando SASL/SCRAM autenticação
- Solução de problemas de conexão

Conectando um cliente ao cluster usando SASL/SCRAM autenticação

1. Execute o comando a seguir em uma máquina que tenha sido AWS CLI instalada. *clusterARN*Substitua pelo ARN do seu cluster.

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

   No resultado JSON desse comando, salve o valor associado à string chamada`BootstrapBrokerStringSaslScram`. Você usará esse valor em etapas posteriores.
2. Em sua máquina cliente, crie um arquivo de configuração JAAS contendo as credenciais de usuário armazenadas em seu segredo. Por exemplo, para o usuário alice, crie um arquivo chamado `users_jaas.conf` com o conteúdo a seguir.

```
KafkaClient {
   org.apache.kafka.common.security.scram.ScramLoginModule required
   username="alice"
   password="alice-secret";
};
```

3. Use o comando a seguir para exportar seu arquivo de configuração JAAS como um parâmetro de ambiente `KAFKA_OPTS`.

```
export KAFKA_OPTS=-Djava.security.auth.login.config=<path-to-jaas-file>/
users_jaas.conf
```

4. Crie um arquivo chamado `kafka.client.truststore.jks` em um diretório `/tmp`.
5. (Opcional) Use o comando a seguir para copiar o arquivo de armazenamento de chaves do JDK da sua `cacerts` pasta JVM para o `kafka.client.truststore.jks` arquivo que você criou na etapa anterior. *JDKFolder*Substitua pelo nome da pasta JDK na sua instância. Por exemplo, sua pasta do JDK pode ter o nome `java-1.8.0-openjdk-1.8.0.201.b09-0.amzn2.x86_64`.

```
cp /usr/lib/jvm/JDKFolder/lib/security/cacerts /tmp/kafka.client.truststore.jks
```

6. No diretório `bin` da instalação do Apache Kafka, crie um arquivo de propriedades do cliente chamado `client_sasl.properties` com o conteúdo a seguir. Esse arquivo define o mecanismo e o protocolo SASL.

```
security.protocol=SASL_SSL
sasl.mechanism=SCRAM-SHA-512
```

7. Para criar um tópico de exemplo, execute o comando a seguir. *BootstrapBrokerStringSaslScram*Substitua pela string do bootstrap broker que você obteve na etapa 1 deste tópico.

```
<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 produzir o tópico de exemplo que você criou, execute o seguinte comando em sua máquina cliente. *BootstrapBrokerStringSaslScram*Substitua pela string do bootstrap broker que você recuperou na etapa 1 deste tópico.

```
<path-to-your-kafka-installation>/bin/kafka-console-producer.sh --broker-
list BootstrapBrokerStringSaslScram --topic ExampleTopicName --producer.config
 client_sasl.properties
```

9. Para consumir do tópico que você criou, execute o comando a seguir em sua máquina cliente. *BootstrapBrokerStringSaslScram*Substitua pela string do bootstrap broker que você obteve na etapa 1 deste tópico.

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBrokerStringSaslScram --topic ExampleTopicName --from-beginning --
consumer.config client_sasl.properties
```

Solução de problemas de conexão

Ao executar comandos do cliente Kafka, você pode encontrar erros de memória de pilha Java, especialmente ao trabalhar com grandes tópicos ou conjuntos de dados. Esses erros ocorrem porque as ferramentas do Kafka são executadas como aplicativos Java com configurações de memória padrão que podem ser insuficientes para sua carga de trabalho.

Para resolver `Out of Memory Java Heap` erros, você pode aumentar o tamanho do heap Java modificando a variável de `KAFKA_OPTS` ambiente para incluir configurações de memória.

O exemplo a seguir define o tamanho máximo da pilha como 1 GB ()`-Xmx1G`. Você pode ajustar esse valor com base na memória e nos requisitos disponíveis do sistema.

```
export KAFKA_OPTS="-Djava.security.auth.login.config=<path-to-jaas-file>/
users_jaas.conf -Xmx1G"
```

Para consumir tópicos extensos, considere usar parâmetros baseados em tempo ou em offset em vez de limitar o uso `--from-beginning` de memória:

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBrokerStringSaslScram --topic ExampleTopicName --max-messages 1000 --
consumer.config client_sasl.properties
```

Trabalhar com usuários

Criação de usuários: você cria usuários como pares de valor/chave em seu segredo. Ao usar a opção Texto simples no console do Secrets Manager, você deve especificar os dados da credencial de login no formato a seguir.

```
{
  "username": "alice",
  "password": "alice-secret"
}
```

Revogando o acesso do usuário: para revogar as credenciais de um usuário para acessar um cluster, recomendamos que primeiro você remova ou force uma ACL no cluster e depois desassocie o segredo. Isso se dá pelo seguinte:

- A remoção de um usuário não fecha as conexões existentes.
- A propagação de alterações em seu segredo levam até 10 minutos.

Para obter informações sobre como usar uma ACL com o Amazon MSK, consulte Apache Kafka ACLs.

Para clusters usando o ZooKeeper modo, recomendamos que você restrinja o acesso aos seus ZooKeeper nós para evitar que os usuários ACLs modifiquem. Para obter mais informações, consulte Controle o acesso aos ZooKeeper nós do Apache em seu cluster Amazon MSK.

Limitações ao usar segredos do SCRAM

Observe as seguintes limitações ao usar segredos SCRAM:

- O Amazon MSK só é compatível com a autenticação SCRAM-SHA-512.
- Um cluster do Amazon MSK pode ter até 1.000 usuários.
- Você deve usar um AWS KMS key com seu segredo. Você não pode usar um segredo que use a chave de criptografia padrão do Secrets Manager com o Amazon MSK. Para obter informações sobre a criação de uma chave do KMS, consulte Criação de chaves do KMS de criptografia simétrica.
- Não é possível usar uma chave assimétrica do KMS com o Secrets Manager.
- Você pode associar até 10 segredos a um cluster por vez usando a BatchAssociateScramSecretoperação.

- O nome dos segredos associados a um cluster do Amazon MSK deve ter o prefixo AmazonMSK_.
- Os segredos associados a um cluster do Amazon MSK devem estar na mesma conta e AWS região da Amazon Web Services do cluster.

Apache Kafka ACLs

O Apache Kafka tem um autorizador conectável e vem com uma implementação autorizadora. out-of-box O Amazon MSK habilita esse autorizador no arquivo `server.properties` dos agentes.

O Apache Kafka ACLs tem o formato “Principal P é [Permitida/Negada] Operação O do Host H em qualquer recurso R correspondente a RP”. ResourcePattern Se RP não corresponder a um recurso específico R, então R não tem nenhum associado e ACLs, portanto, ninguém além de superusuários tem permissão para acessar R. Para alterar esse comportamento do Apache Kafka, você define a propriedade como verdadeira. `allow.everyone.if.no.acl.found` O Amazon MSK a define como true por padrão. Isso significa que, com os clusters do Amazon MSK, se você não definir ACLs explicitamente um recurso, todos os principais poderão acessar esse recurso. Se você ACLs ativar um recurso, somente os diretores autorizados poderão acessá-lo. Se você quiser restringir o acesso a um tópico e autorizar um cliente usando a autenticação mútua TLS, adicione ACLs usando a CLI autorizadora do Apache Kafka. Para obter mais informações sobre adição, remoção e listagem ACLs, consulte Interface de linha de comando de autorização do Kafka.

Como o Amazon MSK configura os agentes como superusuários, eles podem acessar todos os tópicos. Isso ajuda os agentes a replicar mensagens da partição primária, independentemente de a `allow.everyone.if.no.acl.found` propriedade estar definida ou não para a configuração do cluster.

Como adicionar ou remover o acesso de leitura e gravação a um tópico

1. Adicione seus corretores à tabela da ACL para permitir que eles leiam todos os tópicos existentes ACLs . Para conceder acesso de leitura a um tópico para os seus agentes, execute o comando a seguir em uma máquina cliente capaz de se comunicar com o cluster do MSK.

   *`Distinguished-Name`*Substitua pelo DNS de qualquer um dos corretores de bootstrap do seu cluster e, em seguida, substitua a string antes do primeiro ponto nesse nome distinto por um asterisco (). `*` Por exemplo, se um dos corretores de bootstrap do seu cluster tiver o DNS`b-6.mytestcluster.67281x.c4.kafka.us-east-1.amazonaws.com`, *`Distinguished-Name`* substitua o comando a seguir por. `*.mytestcluster.67281x.c4.kafka.us-east-1.amazonaws.com` Para ver informações

sobre como obter os agentes de bootstrap, consulte the section called “Obtenha os corretores de bootstrap”.

```
<path-to-your-kafka-installation>/bin/kafka-acls.sh --bootstrap-server
 BootstrapServerString --add --allow-principal "User:CN=Distinguished-Name" --
operation Read --group=* --topic Topic-Name
```

2. Para conceder a um aplicativo cliente acesso de leitura a um tópico, execute o comando a seguir em sua máquina cliente. Se você usa a autenticação TLS mútua, use a mesma usada *Distinguished-Name* quando criou a chave 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 remover o acesso de leitura, é possível executar o mesmo comando, substituindo `--add` por `--remove`.

3. Para conceder acesso de gravação a um tópico, execute o comando a seguir na máquina de cliente. Se você usa a autenticação TLS mútua, use a mesma usada *Distinguished-Name* quando criou a chave 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 remover o acesso de gravação, é possível executar o mesmo comando, substituindo `--add` por `--remove`.

Alterar o grupo de segurança do cluster no Amazon MSK

Esta página explica como alterar o grupo de segurança de um cluster existente do MSK. Talvez seja necessário alterar o grupo de segurança de um cluster para fornecer acesso a um determinado conjunto de usuários ou limitar o acesso ao cluster. Para obter mais informações sobre grupos de segurança, consulte Grupos de segurança para sua VPC no Guia do usuário da Amazon VPC.

1. Use a ListNodesAPI ou o comando list-nodes no AWS CLI para obter uma lista dos corretores em seu cluster. Os resultados dessa operação incluem as interfaces IDs de rede elástica (ENIs) associadas aos corretores.

2. Faça login no AWS Management Console e abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
3. Usando a lista suspensa no canto superior direito da tela, selecione a região na qual o cluster está implantado.
4. No painel esquerdo, em Rede e Segurança, escolha Interfaces de rede.
5. Selecione a primeira ENI que você obteve na primeira etapa. Escolha o menu Ações na parte superior da tela e escolha Alterar grupos de segurança. Atribua o novo grupo de segurança a essa ENI. Repita essa etapa para cada uma das ENIs que você obteve na primeira etapa.

> **Note**
>
> As alterações que você faz no grupo de segurança de um cluster usando o EC2 console da Amazon não são refletidas no console MSK em Configurações de rede.

6. Configure as regras do novo grupo de segurança para garantir que seus clientes tenham acesso aos agentes. Para obter informações sobre como configurar regras de grupo de segurança, consulte Adicionar, remover e atualizar regras no guia do usuário da Amazon VPC.

> **Important**
>
> Se você alterar o grupo de segurança associado aos agentes de um cluster e depois adicionar novos agentes a esse cluster, o Amazon MSK associará os novos agentes ao grupo de segurança original que estava associado ao cluster quando o cluster foi criado. No entanto, para que um cluster funcione corretamente, todos os seus agentes devem estar associados ao mesmo grupo de segurança. Portanto, se você adicionar novos corretores após alterar o grupo de segurança, deverá seguir novamente o procedimento anterior e atualizar o ENIs dos novos corretores.

Controle o acesso aos ZooKeeper nós do Apache em seu cluster Amazon MSK

Por motivos de segurança, você pode limitar o acesso aos ZooKeeper nós do Apache que fazem parte do seu cluster Amazon MSK. Para limitar o acesso aos nós, é possível atribuir um grupo de segurança separado para eles. Depois, é possível decidir quem tem acesso a esse grupo de segurança.

> **Important**
>
> Esta seção não se aplica a clusters em execução no KRaft modo. Consulte the section called "KRaft modo ".

Este tópico contém as seguintes seções:

Para colocar seus ZooKeeper nós Apache em um grupo de segurança separado

Para limitar o acesso aos ZooKeeper nós do Apache, você pode atribuir um grupo de segurança separado a eles. Você pode escolher quem tem acesso a esse novo grupo de segurança definindo as regras dele.

1. Obtenha a string de ZooKeeper conexão do Apache para seu cluster. Para saber como, consulte the section called "ZooKeeper modo". A string de conexão contém os nomes DNS dos seus nós do Apache ZooKeeper .
2. Use uma ferramenta como `host` ou `ping` para converter os nomes de DNS obtidos na etapa anterior para endereços IP. Salve esses endereços IP porque você precisará deles posteriormente neste procedimento.
3. Faça login no AWS Management Console e abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
4. No painel de navegação, em NETWORK & SECURITY (REDE E SEGURANÇA), selecione Network Interfaces (Interfaces de rede).
5. No campo de pesquisa acima da tabela de interfaces de rede, digite o nome do cluster e digite return. Isso limita o número de interfaces de rede que aparecem na tabela às interfaces associadas ao cluster.
6. Marque a caixa de seleção no início da linha que corresponde à primeira interface de rede na lista.
7. No painel de detalhes na parte inferior da página, procure o IPv4 IP privado primário. Se esse endereço IP corresponder a um dos endereços IP que você obteve na primeira etapa desse procedimento, isso significa que essa interface de rede está atribuída a um ZooKeeper nó Apache que faz parte do seu cluster. Caso contrário, desmarque a caixa de seleção ao lado

dessa interface de rede e selecione a próxima interface de rede na lista. A ordem em que você seleciona as interfaces de rede não importa. Nas próximas etapas, você executará as mesmas operações em todas as interfaces de rede atribuídas aos ZooKeeper nós do Apache, uma por uma.

8. Ao selecionar uma interface de rede que corresponde a um ZooKeeper nó do Apache, escolha o menu Ações na parte superior da página e escolha Alterar grupos de segurança. Atribua um novo grupo de segurança a essa interface de rede. Para obter informações sobre como criar grupos de segurança, consulte Criar um grupo de segurança na documentação da Amazon VPC.
9. Repita a etapa anterior para atribuir o mesmo novo grupo de segurança a todas as interfaces de rede associadas aos ZooKeeper nós Apache do seu cluster.
10. Agora é possível escolher quem tem acesso a esse novo grupo de segurança. Para obter informações sobre como configurar regras de grupo de segurança, consulte Adicionar, remover e atualizar regras na documentação da Amazon VPC.

Usando a segurança TLS com o Apache ZooKeeper

Você pode usar a segurança TLS para criptografia em trânsito entre seus clientes e seus nós Apache ZooKeeper . Para implementar a segurança TLS com seus ZooKeeper nós Apache, faça o seguinte:

- Os clusters devem usar o Apache Kafka versão 2.5.1 ou posterior para usar a segurança TLS com o Apache. ZooKeeper
- Habilite a segurança TLS ao criar ou configurar seu cluster. Clusters criados com o Apache Kafka versão 2.5.1 ou posterior com TLS ativado usam automaticamente a segurança TLS com endpoints Apache. ZooKeeper Para obter mais informações sobre a configuração da segurança TLS, consulte Conceitos básicos sobre criptografia do Amazon MSK.
- Recupere os ZooKeeper endpoints TLS Apache usando a operação. DescribeCluster
- Crie um arquivo de ZooKeeper configuração do Apache para uso com as `kafka-acls.sh`ferramentas `kafka-configs.sh` e ou com o ZooKeeper shell. Com cada ferramenta, você usa o `--zk-tls-config-file` parâmetro para especificar sua ZooKeeper configuração do Apache.

  O exemplo a seguir mostra um arquivo de ZooKeeper configuração típico do 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 outros comandos (como`kafka-topics`), você deve usar a variável de `KAFKA_OPTS` ambiente para configurar ZooKeeper os parâmetros do Apache. O exemplo a seguir mostra como configurar a variável de `KAFKA_OPTS` ambiente para passar ZooKeeper parâmetros do Apache para outros 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"
  ```

  Após configurar a variável de ambiente `KAFKA_OPTS`, você pode usar os comandos da CLI normalmente. O exemplo a seguir cria um tópico do Apache Kafka usando a ZooKeeper configuração do Apache a partir da variável de ambiente: `KAFKA_OPTS`

  ```
  <path-to-your-kafka-installation>/bin/kafka-topics.sh --create --
  zookeeper ZooKeeperTLSConnectString --replication-factor 3 --partitions 1 --topic
   AWSKafkaTutorialTopic
  ```

> **Note**
>
> Os nomes dos parâmetros que você usa no seu arquivo de ZooKeeper configuração do Apache e aqueles que você usa na sua variável de `KAFKA_OPTS` ambiente não são consistentes. Preste atenção nos nomes que você usa com quais parâmetros no arquivo de configuração e na variável de ambiente `KAFKA_OPTS`.

Para obter mais informações sobre como acessar seus ZooKeeper nós Apache com TLS, consulte KIP-515: Habilitar o cliente ZK para usar a nova autenticação compatível com TLS.

Validação de conformidade do Amazon Managed Streaming for Apache Kafka

Auditores terceirizados avaliam a segurança e a conformidade do Amazon Managed Streaming for Apache Kafka como parte de programas de conformidade da AWS . Eles incluem PCI e HIPAA BAA.

Para obter uma lista de AWS serviços no escopo de programas de conformidade específicos, consulte Amazon Services in Scope by Compliance Program . Para obter informações gerais, consulte Programas de AWS conformidade Programas AWS de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixar relatórios em AWS Artifact .

Sua responsabilidade de conformidade ao usar o Amazon MSK é determinada pela confidencialidade dos seus dados, pelos objetivos de conformidade da sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- Guias de início rápido de segurança e compatibilidade: esses guias de implantação abordam as considerações de arquitetura e fornecem etapas para implantação de ambientes de linha de base focados em compatibilidade e segurança na AWS.
- Documento técnico sobre arquitetura para segurança e conformidade com a HIPAA — Este whitepaper descreve como as empresas podem usar para criar aplicativos compatíveis com a HIPAA. AWS
- AWS Recursos de https://aws.amazon.com/compliance/resources/ de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- Avaliação de recursos com regras no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- AWS Security Hub— Esse AWS serviço fornece uma visão abrangente do seu estado de segurança interno, AWS que ajuda você a verificar sua conformidade com os padrões e as melhores práticas do setor de segurança.

Resiliência no Amazon Managed Streaming for Apache Kafka

A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade. AWS As regiões fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte Infraestrutura AWS global.

Segurança de infraestrutura no Amazon Managed Streaming for Apache Kafka

Como um serviço gerenciado, o Amazon Managed Streaming for Apache Kafka é protegido AWS pelos procedimentos globais de segurança de rede descritos no whitepaper Amazon Web Services: Visão geral dos processos de segurança.

Você usa chamadas de API AWS publicadas para acessar o Amazon MSK pela rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.0 ou posterior. Recomendamos TLS 1.2 ou posterior. Os clientes também devem ter compatibilidade com conjuntos de criptografia com perfect forward secrecy (PFS) como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas modernos como Java 7 e versões posteriores oferece compatibilidade com esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou é possível usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Configuração provisionada do Amazon MSK

O Amazon MSK fornece configurações padrão para corretores, tópicos e nós de metadados. Você também pode criar configurações personalizadas e usá-las para criar novos clusters do MSK ou atualizar clusters existentes. Uma configuração do MSK consiste em um conjunto de propriedades e seus valores correspondentes. Dependendo do tipo de agente que você usa em seu cluster, há um conjunto diferente de padrões de configuração e um conjunto diferente de configurações que você pode modificar. Consulte as seções abaixo para obter mais detalhes sobre como configurar seus corretores Standard e Express.

Tópicos

- Configurações padrão de corretor
- Configurações do Express Broker
- Operações de configuração do broker

Configurações padrão de corretor

Esta seção descreve as propriedades de configuração para corretores padrão.

Tópicos

- Configurações personalizadas do Amazon MSK
- Configuração padrão do Amazon MSK
- Diretrizes para a configuração de armazenamento em camadas no nível de tópico do Amazon MSK

Configurações personalizadas do Amazon MSK

É possível usar o Amazon MSK para criar uma configuração personalizada do MSK na qual você define as seguintes propriedades. As propriedades que você não define explicitamente obtêm os valores que têm em the section called “Configuração padrão do Amazon MSK”. Para obter mais informações sobre as propriedades da configuração, consulte Configuração do Apache Kafka.

Propriedades de configuração do Apache Kafka

Nome	Descrição
allow.everyone.if.no.acl.found	Se você quiser definir essa propriedade como`false`, primeiro certifique-se de definir o Apache Kafka ACLs para seu cluster. Se você definir essa propriedade como `false` e não definir primeiro o Apache Kafka ACLs, perderá o acesso ao cluster. Se isso acontecer , é possível atualizar a configuração novamente e definir essa propriedade como `true` para recuperar o acesso ao cluster.
auto.create.topics.enable	Habilita a criação automática de tópicos no servidor.
compression.type	O tipo de compactação final de um determina do tópico. Você pode definir essa proprieda de para os codecs de compactação padrão (`gzip`, `snappy`, `lz4` e `zstd`). Além disso,

Nome	Descrição
	também aceita `uncompressed` . Esse valor é equivalente a nenhuma compactação. Se você definir o valor como `producer`, isso significa reter o codec de compactação original definido pelo produtor.
connections.max.idle.ms	O tempo limite de conexões ociosas em milissegundos. Os threads do processador de soquete do servidor fecham as conexões que estiverem ociosas há mais tempo que o que o valor definido para essa propriedade.
default.replication.factor	O fator de replicação padrão para tópicos criados automaticamente.
delete.topic.enable	Habilita a operação de exclusão de tópico. Se desativar essa configuração, você não poderá excluir um tópico por meio da ferramenta de administração.
group.initial.rebalance.delay.ms	O período que o coordenador do grupo espera que mais consumidores de dados ingressem em um novo grupo antes de executar a primeira operação de rebalanceamento. Um atraso mais longo significa potencialmente menos rebalanceamentos, mas aumenta o tempo até o início do processamento.
group.max.session.timeout.ms	Tempo limite máximo de sessão para consumidores registrados. Tempos limite mais longos permitem que os consumidores tenham mais tempo para processar mensagens entre pulsações ao custo de mais tempo para detectar falhas.

Nome	Descrição
group.min.session.timeout.ms	Tempo limite mínimo de sessão para consumidores registrados. Tempos limite mais curtos resultam em detecção mais rápida de falhas ao custo de pulsações mais frequente s do consumidor. Isso pode sobrecarregar os recursos do agente.
leader.imbalance.per.broker.percentage	A proporção de desequilíbrio de líder permitida por agente. O controlador aciona um balanceamento de líder caso ele ultrapasse esse valor por agente. Esse valor é especific ado em porcentagem.
log.cleaner.delete.retention.ms	Período de tempo que você deseja que o Apache Kafka mantenha registros excluídos. O valor mínimo é 0.
log.cleaner.min.cleanable.ratio	Essa propriedade de configuração pode ter valores entre 0 e 1. Esse valor determina a frequência na qual o compactador de logs tenta limpar o log (se a compactação de logs estiver habilitada). Por padrão, o Apache Kafka evita limpar um log se mais de 50% do log tiver sido compactado. Essa proporção limita o espaço máximo que o log desperdiça com duplicata s (em 50%, isso significa que até 50% do log pode ser de duplicatas). Uma proporção maior significa menos limpezas mais eficientes, mas também mais espaço desperdiçado no log.

Nome	Descrição
log.cleanup.policy	A política de limpeza padrão para segmentos além da janela de retenção. Uma lista de políticas válidas separadas por vírgulas. As políticas válidas são `delete` e `compact`. Para clusters habilitados para armazenamento em camadas, a política válida é somente `delete`.
log.flush.interval.messages	O número de mensagens acumuladas em uma partição de log antes que as mensagens sejam liberadas para o disco.
log.flush.interval.ms	O período máximo em milissegundos no qual uma mensagem em qualquer tópico permanece na memória antes de ser liberada para o disco. Se você não definir esse valor, o sistema usará o valor em log.flush.schedule r.interval.ms. O valor mínimo é 0.
log.message.timestamp.difference.max.ms	Essa configuração está obsoleta no Kafka 3.6.0. Duas configurações, `log.messa ge.timestamp.before.max.ms` `elog.message.timestamp.after .max.ms` , foram adicionadas. A diferença máxima de tempo entre o carimbo de data/hora em que um agente recebe uma mensagem e o carimbo de data/hora especificado na mensagem. Se log.message.timestamp.type= CreateTime, uma mensagem será rejeitada se a diferença no timestamp exceder esse limite. Essa configuração será ignorada se log.messa ge.timestamp.type=. LogAppendTime

Nome	Descrição
log.message.timestamp.type	Especifica se o carimbo de data/hora na mensagem é o horário de criação da mensagem ou da adição no log. Os valores permitidos são `CreateTime` e `LogAppendTime`.
log.retention.bytes	Tamanho máximo do log antes de ser excluído.
log.retention.hours	Número de horas para manter um arquivo de log antes de excluí-lo, terciário à propriedade log.retention.ms.
log.retention.minutes	Número de minutos para manter um arquivo de log antes de excluí-lo, secundário à proprieda de log.retention.ms. Se você não definir esse valor, o sistema usará o valor de log.reten tion.hours.
log.retention.ms	Número de milissegundos para manter um arquivo de log antes de excluí-lo (em milissegu ndos). Se não for definido, o valor de log.reten tion.minutes será usado.
log.roll.ms	Tempo máximo para que um novo segmento de log seja implantado (em milissegundos). Se você não definir essa propriedade, o sistema usará o valor de log.roll.hours. O valor mínimo possível para essa propriedade é 1.
log.segment.bytes	Tamanho máximo de um único arquivo de log.
max.incremental.fetch.session.cache.slots	Número máximo de sessões de busca incrementais mantidas.

Nome	Descrição
message.max.bytes	O maior tamanho de lote de registros que o Kafka permite. Se você aumentar esse valor e houver consumidores anteriores à versão 0.10.2, também será necessário aumentar o tamanho de busca dos consumidores para que eles possam buscar lotes de registros desse tamanho. O formato de mensagem mais recente sempre agrupa as mensagens em lotes visando eficiência. As versões anteriores de formato de mensagem não agrupam em lotes os registros não compactados, e, nesse caso, esse limite é aplicável somente a um único registro. É possível definir esse valor por tópico com a configuração max.message.bytes de nível do tópico.

Nome	Descrição
min.insync.replicas	Quando um produtor define acks como `"all"` (ou `"-1"`), o valor em min.insync.replicas especifica o número mínimo de réplicas que devem confirmar uma gravação para que a gravação seja considerada bem-sucedida. Se esse mínimo não puder ser atingido, o produtor cria uma exceção (NotEnoughReplicas ou NotEnoughReplicasAfterAppend). Você pode usar valores em min.insync.replica s e acks para forçar maiores garantias de durabilidade. Por exemplo, você poderia criar um tópico com um fator de replicação de 3, definir min.insync.replicas como 2 e produzir com acks de `"all"`. Isso garante que o produtor gere uma exceção se a maioria das réplicas não receber uma gravação.
num.io.threads	O número de threads que o servidor usa para processar solicitações, que podem incluir E/S de disco.
num.network.threads	O número de threads que o servidor usa para receber solicitações da rede e enviar respostas para ela.
num.partitions	Número padrão de partições de log por tópico.
num.recovery.threads.per.data.dir	O número de threads por diretório de dados a ser usado para recuperar logs na inicialização e para liberá-los no desligamento.
num.replica.fetchers	O número de threads de busca usados para replicar mensagens de um agente de origem. Se você aumentar esse valor, poderá aumentar o grau de I/O paralelismo no corretor seguidor.

Nome	Descrição
offsets.retention.minutes	Depois que um grupo de consumidores perde todos os consumidores (isto é, torna-se vazio), seus deslocamentos são mantidos durante esse período de retenção antes de serem descartados. Para consumidores autônomos (ou seja, que usam atribuição manual), os deslocamentos expiram depois da última confirmação somada a esse período de retenção.
offsets.topic.replication.factor	O fator de replicação do tópico de deslocamento. Defina esse valor mais alto para garantir a disponibilidade. A criação do tópico interno falha até que o tamanho do cluster atenda a esse requisito de fator de replicação.
replica.fetch.max.bytes	O número de bytes de mensagens para tentar buscar para cada partição. Esse não é um máximo absoluto. Se o primeiro lote de registros na primeira partição não vazia da busca for maior que esse valor, o lote de registros será retornado para garantir o progresso. As propriedades message.max.bytes (configuração do agente) ou max.message.bytes (configuração do tópico) definem o tamanho máximo do lote de registros aceito pelo agente.

Nome	Descrição
replica.fetch.response.max.bytes	O número máximo de bytes esperado para toda a resposta de busca. Os registros são buscados em lotes e, se o primeiro lote de registros na primeira partição não vazia da busca for maior que esse valor, o lote de registros ainda será retornado para garantir o progresso. Esse não é um máximo absoluto. As propriedades message.max.bytes (configuração do agente) ou max.message.bytes (configuração do tópico) especificam o tamanho máximo do lote de registros aceito pelo agente.
replica.lag.time.max.ms	Se um seguidor não enviou nenhuma solicitação de busca ou se não consumiu até o deslocamento final do log do líder por pelo menos esse número de milissegundos, o líder remove o seguidor do ISR. MinValue: 10000 MaxValue = 30000
replica.selector.class	O nome da classe totalmente qualificado que implementa. ReplicaSelector O agente usa esse valor para encontrar a réplica de leitura preferencial. Se estiver usando a versão 2.4.1 ou mais recente do Apache Kafka e quiser permitir que os clientes busquem da réplica mais próxima, defina essa propriedade como `org.apache.kafka.common.replica.RackAwareReplicaSelector`. Para obter mais informações, consulte the section called “Apache Kafka versão 2.4.1 (use 2.4.1.1 alternativamente)”.

Nome	Descrição
replica.socket.receive.buffer.bytes	O buffer de recebimento do soquete para solicitações de rede.
socket.receive.buffer.bytes	O buffer SO_RCVBUF dos soquetes do servidor de soquetes. O valor mínimo que você pode definir para essa propriedade é -1. Se o valor for -1, o Amazon MSK usará o sistema operacional padrão.
socket.request.max.bytes	O número máximo de bytes em uma solicitação de soquete.
socket.send.buffer.bytes	O buffer SO_SNDBUF dos soquetes do servidor de soquetes. O valor mínimo que você pode definir para essa propriedade é -1. Se o valor for -1, o Amazon MSK usará o sistema operacional padrão.
transaction.max.timeout.ms	Tempo limite máximo para transações. Se o tempo de transação solicitado de um cliente exceder esse valor, o corretor retornará um erro em InitProducerIdRequest. Isso impede que um cliente use um tempo limite muito grande e pode impedir que os consumidores leiam os tópicos incluídos na transação.
transaction.state.log.min.isr	A configuração de min.insync.replicas substituída para o tópico de transação.
transaction.state.log.replication.factor	O fator de replicação do tópico de transação. Defina essa propriedade com um valor maior para aumentar a disponibilidade. A criação do tópico interno falha até que o tamanho do cluster atenda a esse requisito de fator de replicação.

Nome	Descrição
transactional.id.expiration.ms	O tempo em milissegundos que o coordenador da transação deve aguardar para receber qualquer atualização do status da transação atual antes que o coordenador expire sua ID transacional. Essa configuração também influencia a expiração do ID do produtor porque faz com que o produtor IDs expire quando esse tempo decorre após a última gravação com o ID do produtor fornecido. O produtor IDs pode expirar mais cedo se a última gravação do ID do produtor for excluída devido às configurações de retenção do tópico. O valor mínimo para essa propriedade é de 1 milissegundo.
unclean.leader.election.enable	Indica se as réplicas que não estão no conjunto ISR devem atuar como líderes em último recurso, mesmo que isso possa resultar em perda de dados.
zookeeper.connection.timeout.ms	ZooKeeper clusters de modos. Tempo máximo que o cliente espera para estabelecer uma conexão. ZooKeeper Se você não definir esse valor, o sistema usará o valor de zookeeper.session.timeout.ms. MinValue = 6000 MaxValue (inclusive) = 18000 Recomendamos que você defina esse valor como 10.000 em T3.small para evitar o tempo de inatividade do cluster.

Nome	Descrição
zookeeper.session.timeout.ms	ZooKeeper clusters de modos. O tempo limite da ZooKeeper sessão do Apache em milissegundos. MinValue = 6000 MaxValue (inclusive) = 18000

Para saber como criar uma configuração personalizada do MSK, listar todas as configurações ou descrevê-las, consulte the section called “Operações de configuração do broker”. Para criar um cluster do MSK com uma configuração personalizada do MSK ou para atualizar um cluster com uma nova configuração personalizada, consulte the section called “Principais características e conceitos”.

Quando você atualiza o cluster existente do MSK com uma configuração personalizada do MSK, o Amazon MSK faz reinicializações contínuas quando necessário, empregando as práticas recomendadas para minimizar o tempo de inatividade do cliente. Por exemplo, depois que o Amazon MSK reinicia cada agente, o Amazon MSK tenta deixar o agente recuperar os dados que possam ter sido perdidos pelo agente durante a atualização da configuração antes de avançar para o próximo agente.

Configuração dinâmica do Amazon MSK

Além das propriedades de configuração fornecidas pelo Amazon MSK, você pode definir dinamicamente as propriedades de configuração em nível de cluster e de agente que não exigem uma reinicialização do agente. É possível definir dinamicamente algumas propriedades de configuração. Trata-se das propriedades que não estão marcadas como somente leitura na tabela em Configurações do agente na documentação do Apache Kafka. Para obter informações sobre a configuração dinâmica e comandos de exemplo, consulte Atualização das configurações do agente na documentação do Apache Kafka.

> **Note**
>
> É possível definir a propriedade `advertised.listeners`, mas não a propriedade `listeners`.

Configuração no nível de tópico do Amazon MSK

Você pode usar os comandos do Apache Kafka para definir ou modificar propriedades de configuração em nível de tópico para tópicos novos e existentes. Para obter mais informações sobre as propriedades de configuração no nível de tópico e exemplos sobre como defini-las, consulte Configurações no nível de tópico na documentação do Apache Kafka.

Configuração padrão do Amazon MSK

Quando você cria um cluster do MSK sem especificar uma configuração personalizada do MSK, o Amazon MSK cria e usa uma configuração padrão com os valores apresentados na tabela a seguir. Para propriedades que não estejam nessa tabela, o Amazon MSK usará os padrões associados à sua versão do Apache Kafka. Para obter uma lista desses valores padrão, consulte Configuração do Apache Kafka.

Valores padrão de configuração

Nome	Descrição	Valor padrão para cluster de armazenamento sem camadas	Valor padrão para cluster de armazenamento em camadas
allow.everyone.if.no.acl.found	Se nenhum padrão de recurso corresponder a um recurso específico, o recurso não tem nenhum associado ACLs. Nesse caso, se você definir essa propriedade como `true`, todos os usuários terão acesso ao recurso, não apenas os superusuários.	`true`	`true`
auto.create.topics.enable	Habilita a criação automática de um tópico no servidor.	`false`	`false`

Nome	Descrição	Valor padrão para cluster de armazenamento sem camadas	Valor padrão para cluster de armazenamento em camadas
auto.leader.rebalance.enable	Habilita o equilíbrio de líderes automáticos. Se necessário, um thread em segundo plano verifica e inicia o balanceamento do líder em intervalos regulares.	`true`	`true`
default.replication.factor	Fatores de replicação padrão para tópicos criados automaticamente.	O valor é 3 para clusters em 3 zonas de disponibilidade e 2 para clusters em 2 zonas de disponibilidade.	O valor é 3 para clusters em 3 zonas de disponibilidade e 2 para clusters em 2 zonas de disponibilidade.

Nome	Descrição	Valor padrão para cluster de armazenamento sem camadas	Valor padrão para cluster de armazenamento em camadas
local.retention.bytes	O tamanho máximo dos segmentos de log locais de uma partição antes que ela exclua os segmentos antigos. Se você não definir esse valor, o sistema usará o valor de log.retention.bytes. O valor efetivo sempre deve ser menor que ou igual ao valor de log.retention.bytes. O valor padrão de -2 indica que não há limite para a retenção local. Isso corresponde à configuração de -1 para retention.ms/bytes. As propriedades local.retention.ms e local.retention.bytes são semelhantes a log.retention, pois são usadas para determinar por quanto tempo os segmentos de log devem permanecer no armazenamento local. As configura	-2 para ilimitado	-2 para ilimitado

Nome	Descrição	Valor padrão para cluster de armazenamento sem camadas	Valor padrão para cluster de armazenamento em camadas
	ções existentes de log.retention.* são configurações de retenção para a partição do tópico. Isso inclui armazenamento local e remoto. Valores válidos: números inteiros em [-2; +Inf]		

Nome	Descrição	Valor padrão para cluster de armazenamento sem camadas	Valor padrão para cluster de armazenamento em camadas
local.retention.ms	O número de milissegundos para a retenção do segmento de log local antes da exclusão. Se você não definir esse valor, o Amazon MSK usará o valor de log.retention.ms. O valor efetivo sempre deve ser menor que ou igual ao valor de log.retention.bytes. O valor padrão de -2 indica que não há limite para a retenção local. Isso corresponde à configuração de -1 para retention.ms/bytes. Os valores de local.retention.ms e local.retention.bytes são semelhantes a log.retention. O MSK usa essa configuração para determinar por quanto tempo os segmentos de log devem permanecer no armazenamento local. As configura	-2 para ilimitado	-2 para ilimitado

Nome	Descrição	Valor padrão para cluster de armazenamento sem camadas	Valor padrão para cluster de armazenamento em camadas
	ções existentes de log.retention.* são configurações de retenção para a partição do tópico. Isso inclui armazenamento local e remoto. Os valores válidos são números inteiros maiores que 0.		

Nome	Descrição	Valor padrão para cluster de armazenamento sem camadas	Valor padrão para cluster de armazenamento em camadas
log.message.timestamp.difference.max.ms	Essa configuração está obsoleta no Kafka 3.6.0. Duas configurações, `log.message.timestamp.before.max.ms` `elog.message.timestamp.after.max.ms` , foram adicionadas. A diferença máxima permitida entre o timestamp em que um agente recebe uma mensagem e o timestamp especificado na mensagem. Se log.message.timestamp.type=CreateTime, uma mensagem será rejeitada se a diferença no timestamp exceder esse limite. Essa configuração será ignorada se log.message.timestamp.type=.	9223372036854775807	86400000 para Kafka 2.8.2.tiered e Kafka 3.7.x em camadas.

Nome	Descrição	Valor padrão para cluster de armazenamento sem camadas	Valor padrão para cluster de armazenamento em camadas
	LogAppendTime Para evitar a repetição desnecessária e frequente de registros, a diferença máxima permitida para o carimbo de data/hora não deve ser maior que log.retention.ms.		
log.segment.bytes	O tamanho máximo de um único arquivo de log.	1073741824	134217728

Nome	Descrição	Valor padrão para cluster de armazenamento sem camadas	Valor padrão para cluster de armazenamento em camadas
min.insync.replicas	Quando um produtor define o valor de confirmações (as confirmações que o produtor receber o agente do Kafka) como `"all"` (ou `"-1"`), o valor em min.insync.replicas especifica o número mínimo de réplicas que devem confirmar uma gravação para que a gravação seja considerada bem-sucedida. Se esse valor não atingir esse mínimo, o produtor gera uma exceção (NotEnoughReplicas ou NotEnough ReplicasAfterAppend). Quando você usar os valores em min.insync.replicas e acks juntos, será possível forçar maiores garantias de durabilidade. Por exemplo, você poderia criar um tópico com um	O valor é 2 para clusters em 3 zonas de disponibilidade e 1 para clusters em 2 zonas de disponibilidade.	O valor é 2 para clusters em 3 zonas de disponibilidade e 1 para clusters em 2 zonas de disponibilidade.

Nome	Descrição	Valor padrão para cluster de armazenamento sem camadas	Valor padrão para cluster de armazenamento em camadas
	fator de replicação de 3, definir min.insync.replicas como 2 e produzir com acks de `"all"`. Isso garante que o produtor gere uma exceção se a maioria das réplicas não receber uma gravação.		
num.io.threads	O número de threads que o servidor usa para produzir solicitações, que podem incluir E/S de disco.	8	max (8, vCPUs) onde v CPUs depende do tamanho da instância do broker
num.network.threads	O número de threads que o servidor usa para receber solicitações da rede e enviar respostas para a rede.	5	max (5, vCPUs /2) onde v CPUs depende do tamanho da instância do broker
num.partitions	Número padrão de partições de log por tópico.	1	1

Nome	Descrição	Valor padrão para cluster de armazenamento sem camadas	Valor padrão para cluster de armazenamento em camadas
num.replica.fetchers	Número de segmentos de busca usados para replicar mensagens de um corretor de origem. Se você aumentar esse valor, poderá aumentar o grau de I/O paralelismo no corretor seguidor.	2	max (2, vCPUs /4) onde v CPUs depende do tamanho da instância do broker
remote.log.msk.disable.policy	Usado com remote.storage.enable para desabilitar o armazenamento em camadas. Defina essa política como Excluir para indicar que os dados no armazenamento em camadas são excluídos quando você definir remote.storage.enable como falso.	N/D	Nenhum
remote.log.reader.threads	O tamanho do pool de threads do leitor de logs remoto. Usado no agendamento de tarefas para buscar dados do armazenamento remoto.	N/D	max (10, v CPUs * 0,67) onde v CPUs depende do tamanho da instância do broker

Nome	Descrição	Valor padrão para cluster de armazenamento sem camadas	Valor padrão para cluster de armazenamento em camadas
remote.storage.enable	Se definido como verdadeiro, habilita o armazenamento em camadas (remoto) para um tópico. Desabilita o armazenamento em camadas no nível de tópico se definido como falso e se remote.log.msk.disable.policy estiver definido como Excluir. Ao desabilitar o armazenamento em camadas, você exclui dados do armazenamento remoto. Ao desabilitar o armazenamento em camadas para um tópico, não será possível habilitá-lo novamente.	false	false

Nome	Descrição	Valor padrão para cluster de armazenamento sem camadas	Valor padrão para cluster de armazenamento em camadas
replica.lag.time.max.ms	Se um seguidor não enviou nenhuma solicitação de busca ou se não consumiu até o deslocamento final do log do líder por pelo menos esse número de milissegundos, o líder remove o seguidor do ISR.	30000	30000

Nome	Descrição	Valor padrão para cluster de armazenamento sem camadas	Valor padrão para cluster de armazenamento em camadas
retention.ms	Campo obrigatório. O tempo mínimo é de 3 dias. Não há padrão porque a configuração é obrigatória. O Amazon MSK usa o valor retention.ms com local.retention.ms para determinar quando os dados são movidos do armazenamento local para o armazenamento em camadas. O valor local.retention.ms especifica quando mover dados do armazenamento local para o armazenamento em camadas. O valor retention.ms especifica quando remover dados do armazenamento em camadas (ou seja, remoção do cluster). Valores válidos: números inteiros em [-1; +Inf]	Mínimo de 259.200.000 milissegundos (3 dias). Use -1 para retenção infinita.	Mínimo de 259.200.000 milissegundos (3 dias). Use -1 para retenção infinita.

Nome	Descrição	Valor padrão para cluster de armazenamento sem camadas	Valor padrão para cluster de armazenamento em camadas
socket.receive.buffer.bytes	O buffer SO_RCVBUF dos soquetes do servidor de soquetes. Se o valor for -1, o sistema operacional padrão será usado.	102400	102400
socket.request.max.bytes	Número máximo de bytes em uma solicitação de soquete.	104857600	104857600
socket.send.buffer.bytes	O buffer SO_SNDBUF dos soquetes do servidor de soquetes. Se o valor for -1, o sistema operacional padrão será usado.	102400	102400
unclean.leader.election.enable	Indica se você deseja que as réplicas que não estão no conjunto ISR devem atuar como líderes em último recurso, mesmo que isso possa resultar em perda de dados.	verdadeiro	false
zookeeper.session.timeout.ms	O tempo limite da ZooKeeper sessão do Apache em milissegundos.	18000	18000

Nome	Descrição	Valor padrão para cluster de armazenamento sem camadas	Valor padrão para cluster de armazenamento em camadas
zookeeper.set.acl	O cliente definido a ser usado com segurança ACLs.	false	false

Para obter mais informações sobre como definir valores de configuração personalizada, consulte the section called “Configurações personalizadas do Amazon MSK”.

Diretrizes para a configuração de armazenamento em camadas no nível de tópico do Amazon MSK

Veja a seguir as configurações e limitações padrão quando você configura o armazenamento em camadas no nível de tópico.

- O Amazon MSK não é compatível com tamanhos menores de segmentos de log para tópicos com o armazenamento em camadas ativado. Se você quiser criar um segmento, há um tamanho mínimo de segmento de log de 48 MiB ou um tempo mínimo de rolagem do segmento de 10 minutos. Esses valores são mapeados para as propriedades segment.bytes e segment.ms.
- O valor de local.retention. ms/bytes can't equal or exceed the retention.ms/bytes. Essa é a configuração de retenção de armazenamento em camadas.
- O valor padrão para local.retention. ms/bytes is -2. This means that the retention.ms value is used for local.retention.ms/bytes. Nesse caso, os dados permanecem no armazenamento local e no armazenamento em camadas (uma cópia em cada) e expiram juntos. Para essa opção, uma cópia dos dados locais é mantida no armazenamento remoto. Nesse caso, os dados lidos do tráfego de consumo vêm do armazenamento local.
- O valor padrão para retention.ms é de 7 dias. Não há limite de tamanho padrão para retention.bytes.
- O valor mínimo para retention.ms/bytes é -1. Isso significa retenção infinita.
- O valor mínimo para local.retention. ms/bytes is -2. This means infinite retention for local storage. It matches with the retention.ms/bytesdefinindo como -1.
- A configuração retention.ms no nível de tópico é obrigatória para tópicos com armazenamento em camadas ativado. O mínimo de retention.ms é de 3 dias.

Para obter mais informações sobre restrições de armazenamento hierárquico, consulte. Restrições e limitações do armazenamento em camadas para clusters do Amazon MSK

Configurações do Express Broker

O Apache Kafka tem centenas de configurações de agente que você pode usar para ajustar o desempenho do seu cluster provisionado pelo MSK. Definir valores errôneos ou abaixo do ideal pode afetar a confiabilidade e o desempenho do cluster. Os corretores expressos melhoram a disponibilidade e a durabilidade de seus clusters provisionados pela MSK definindo valores ideais para configurações críticas e protegendo-os de configurações incorretas comuns. Há três categorias de configurações com base no acesso de leitura e gravação: configurações de leitura/gravação (editável), somente leitura e configurações de não leitura/gravação. Algumas configurações ainda usam o valor padrão do Apache Kafka para a versão do Apache Kafka que o cluster está executando. Nós os marcamos como Apache Kafka Default.

Tópicos

- Configurações personalizadas do agente MSK Express (acesso de leitura/gravação)
- Configurações somente para leitura do Express Brokers

Configurações personalizadas do agente MSK Express (acesso de leitura/gravação)

Você pode atualizar as configurações do read/write broker usando o recurso de configuração de atualização do Amazon MSK ou usando a API do Apache Kafka. AlterConfig As configurações do corretor Apache Kafka são estáticas ou dinâmicas. As configurações estáticas exigem a reinicialização do broker para que a configuração seja aplicada, enquanto as configurações dinâmicas não precisam da reinicialização do broker. Para obter mais informações sobre propriedades de configuração e modos de atualização, consulte Atualização das configurações do broker.

Tópicos

- Configurações estáticas em corretores MSK Express
- Configurações dinâmicas em Express Brokers
- Configurações em nível de tópico em Express Brokers

Configurações estáticas em corretores MSK Express

Você pode usar o Amazon MSK para criar um arquivo de configuração MSK personalizado para definir as seguintes propriedades estáticas. O Amazon MSK define e gerencia todas as outras propriedades que você não configura. Você pode criar e atualizar arquivos de configuração estáticos no console MSK ou usando o comando configurations.

Propriedade	Descrição	Valor padrão
allow.everyone.if.no.acl.found	Se você quiser definir essa propriedade como false, primeiro certifique-se de definir o Apache Kafka ACLs para seu cluster. Se você definir essa propriedade como false e não definir primeiro o Apache Kafka ACLs, perderá o acesso ao cluster. Se isso acontecer, você poderá atualizar a configuração novamente e definir essa propriedade como verdadeir a para recuperar o acesso ao cluster.	true
auto.create.topics.enable	Habilita a criação automática de um tópico no servidor.	false
compression.type	Especifique o tipo de compactação final para um determinado tópico. Essa configuração aceita os codecs de compactação padrão: gzip, snappy, lz4, zstd. Essa configuração também aceita`uncompressed` , o que equivale a nenhuma	Apache Kafka padrão

Propriedade	Descrição	Valor padrão
	compactação; e`producer`, o que significa reter o codec de compactação original definido pelo produtor.	
connections.max.idle.ms	O tempo limite de conexões ociosas em milissegundos. Os threads do processador de soquete do servidor fecham as conexões que estiverem ociosas há mais tempo que o que o valor definido para essa propriedade.	Apache Kafka padrão
delete.topic.enable	Habilita a operação de exclusão de tópico. Se desativar essa configuração, você não poderá excluir um tópico por meio da ferramenta de administração.	Apache Kafka padrão
group.initial.rebalance.delay.ms	O período que o coordenador do grupo espera que mais consumidores de dados ingressem em um novo grupo antes de executar a primeira operação de rebalanceamento. Um atraso mais longo significa potencialmente menos rebalanceamentos, mas aumenta o tempo até o início do processamento.	Apache Kafka padrão

Propriedade	Descrição	Valor padrão
group.max.session.timeout.ms	Tempo limite máximo de sessão para consumidores registrados. Tempos limite mais longos permitem que os consumidores tenham mais tempo para processar mensagens entre pulsações ao custo de mais tempo para detectar falhas.	Apache Kafka padrão
leader.imbalance.per.broker .percentage	A proporção de desequilíbrio de líder permitida por agente. O controlador aciona um balanceamento de líder caso ele ultrapasse esse valor por agente. Esse valor é especific ado em porcentagem.	Apache Kafka padrão
log.cleanup.policy	A política de limpeza padrão para segmentos além da janela de retenção. Uma lista de políticas válidas separadas por vírgulas. As políticas válidas são `delete` e `compact`. Para clusters habilitados para armazenam ento hierárquico, a política válida é somente. `delete`	Apache Kafka padrão

Propriedade	Descrição	Valor padrão
log.message.timestamp.after.max.ms	A diferença de data e hora permitida entre o timestamp da mensagem e o timestamp do corretor. O timestamp da mensagem pode ser posterior ou igual ao timestamp do broker, com a diferença máxima permitida determinada pelo valor definido nessa configuração. Se `log.message.timestamp.type=CreateTime`, a mensagem será rejeitada se a diferença nos timestamps exceder esse limite especificado. Essa configuração é ignorada se `log.message.timestamp.type=LogAppendTime`.	86400000 (24 * 60 * 60 * 1000 ms, ou seja, 1 dia)

Propriedade	Descrição	Valor padrão
log.message.timestamp.before.max.ms	A diferença de data e hora permitida entre o timestamp do broker e o timestamp da mensagem. O timestamp da mensagem pode ser anterior ou igual ao timestamp do broker, com a diferença máxima permitida determinada pelo valor definido nessa configuração. Se `log.message.timestamp.type=CreateTime`, a mensagem será rejeitada se a diferença nos timestamps exceder esse limite especificado. Essa configuração é ignorada se `log.message.timestamp.type=LogAppendTime`.	86400000 (24 * 60 * 60 * 1000 ms, ou seja, 1 dia)
log.message.timestamp.type	Especifica se o carimbo de data/hora na mensagem é o horário de criação da mensagem ou da adição no log. Os valores permitidos são `CreateTime` e `LogAppendTime`.	Apache Kafka padrão
log.retention.bytes	Tamanho máximo do log antes de ser excluído.	Apache Kafka padrão
log.retention.ms	Número de milissegundos para manter um arquivo de log antes de excluí-lo.	Apache Kafka padrão

Propriedade	Descrição	Valor padrão
max.connections.por ip	O número máximo de conexões permitidas de cada endereço IP. Isso pode ser definido como `0` se houver substituições configuradas usando a `max.connections.per.ip.overrides` propriedade. Novas conexões do endereço IP são eliminadas se o limite for atingido.	Apache Kafka padrão
max.incremental.fetch.session.cache.slots	Número máximo de sessões de busca incrementais mantidas.	Apache Kafka padrão

Propriedade	Descrição	Valor padrão
message.max.bytes	O maior tamanho de lote de registros que o Kafka permite. Se você aumentar esse valor e houver consumidores anteriores à versão 0.10.2, também será necessário aumentar o tamanho de busca dos consumidores para que eles possam buscar lotes de registros desse tamanho. O formato de mensagem mais recente sempre agrupa as mensagens em lotes visando eficiência. As versões anteriores de formato de mensagem não agrupam em lotes os registros não compactados, e, nesse caso, esse limite é aplicável somente a um único registro. Você pode definir esse valor por tópico com a `max.message.bytes` configuração do nível do tópico.	Apache Kafka padrão
num.partitions	Número padrão de partições por tópico.	1

Propriedade	Descrição	Valor padrão
offsets.retention.minutes	Depois que um grupo de consumidores perde todos os consumidores (isto é, torna-se vazio), seus deslocamentos são mantidos durante esse período de retenção antes de serem descartados. Para consumidores autônomos (ou seja, aqueles que usam atribuição manual), as compensações expiram após a data da última confirmaç ão mais esse período de retenção.	Apache Kafka padrão
replica.fetch.max.bytes	O número de bytes de mensagens para tentar buscar para cada partição. Esse não é um máximo absoluto. Se o primeiro lote de registros na primeira partição não vazia da busca for maior que esse valor, o lote de registros será retornado para garantir o progresso. As proprieda des message.max.bytes (configuração do agente) ou max.message.bytes (configur ação do tópico) definem o tamanho máximo do lote de registros aceito pelo agente.	Apache Kafka padrão

Propriedade	Descrição	Valor padrão
replica.selector.class	O nome da classe totalmente qualificado que implementa. ReplicaSelector O agente usa esse valor para encontrar a réplica de leitura preferencial. Se você quiser permitir que os consumidores busquem na réplica mais próxima, defina essa propriedade como. `org.apache.kafka.common.replica.RackAwareReplicaSelector`	Apache Kafka padrão
socket.receive.buffer.bytes	O buffer SO_RCVBUF dos soquetes do servidor de soquetes. Se o valor for -1, o sistema operacional padrão será usado.	102400
socket.request.max.bytes	Número máximo de bytes em uma solicitação de soquete.	104857600
socket.send.buffer.bytes	O buffer SO_SNDBUF dos soquetes do servidor de soquetes. Se o valor for -1, o sistema operacional padrão será usado.	102400

Propriedade	Descrição	Valor padrão
transaction.max.timeout.ms	Tempo limite máximo para transações. Se o tempo de transação solicitado de um cliente exceder esse valor, o corretor retornará um erro em InitProducerIdRequest. Isso impede que um cliente use um tempo limite muito grande e pode impedir que os consumidores leiam os tópicos incluídos na transação.	Apache Kafka padrão
transactional.id.expiration.ms	O tempo em milissegu ndos que o coordenador da transação deve aguardar para receber qualquer atualizaç ão do status da transação atual antes que o coordenad or expire sua ID transacional. Essa configuração também influencia a expiração do ID do produtor porque faz com que IDs o produtor expire quando esse tempo decorre após a última gravação com o ID do produtor fornecido. O produtor IDs pode expirar mais cedo se a última gravação do ID do produtor for excluída devido às configura ções de retenção do tópico. O valor mínimo para essa propriedade é de 1 milissegu ndo.	Apache Kafka padrão

Configurações dinâmicas em Express Brokers

Você pode usar a AlterConfig API Apache Kafka ou a ferramenta Kafka-configs.sh para editar as seguintes configurações dinâmicas. O Amazon MSK define e gerencia todas as outras propriedades que você não configura. Você pode definir dinamicamente propriedades de configuração em nível de cluster e em nível de intermediário que não exijam a reinicialização do agente.

Propriedade	Descrição	Valor padrão
ouvintes anunciados	Ouvintes a serem publicados para os clientes usarem, se forem diferentes da propriedade de `listeners` configuração. Em ambientes de IaaS, isso pode precisar ser diferente da interface à qual o corretor se vincula. Se isso não for definido, o valor para ouvintes será usado. Ao contrário dos ouvintes, não é válido anunciar o meta-ende reço 0.0.0.0.	nulo

Propriedade	Descrição	Valor padrão
	Além dis`listeners`, pode haver portas duplicadas nessa propriedade, de forma que um ouvinte possa ser configurado para anunciar o endereço de outro ouvinte. Isso pode ser útil em alguns casos em que balanceadores de carga externos são usados. Essa propriedade é definida em um nível por corretor.	

Propriedade	Descrição	Valor padrão
compression.type	O tipo de compactação final de um determinado tópico. Você pode definir essa propriedade para os codecs de compactação padrão (`gzip`, `snappy`, `lz4` e `zstd`). Além disso, também aceita `uncompressed` . Esse valor é equivalente a nenhuma compactação. Se você definir o valor como `producer`, isso significa reter o codec de compactação original definido pelo produtor.	Apache Kafka padrão

Propriedade	Descrição	Valor padrão
log.cleaner.delete.retention.ms	A quantidade de tempo necessário para reter marcadores de lápide excluídos para tópicos compactados de registros. Essa configuração também fornece um limite no tempo em que um consumidor deve concluir uma leitura se começar do deslocamento 0 para garantir que obtenha um instantâneo válido do estágio final. Caso contrário, as lápides excluídas podem ser coletadas antes que elas concluam a digitalização.	86400000 (24 * 60 * 60 * 1000 ms, ou seja, 1 dia), Apache Kafka Default

Propriedade	Descrição	Valor padrão
log.cleaner.min.compaction.lag.ms	O tempo mínimo em que uma mensagem permanecerá descompactada no registro. Essa configuração é aplicável somente para registros que estão sendo compactados.	0, Apache Kafka padrão

Propriedade	Descrição	Valor padrão
log.cleaner.max.compaction.lag.ms	O tempo máximo em que uma mensagem permanecerá inelegível para compactação no registro. Essa configuração é aplicável somente para registros que estão sendo compactados. Essa configuração seria limitada no intervalo de [7 dias, Long.Max].	9223372036854775807, Apache Kafka padrão

Propriedade	Descrição	Valor padrão
log.clean up.policy	A política de limpeza padrão para segmentos além da janela de retenção. Uma lista de políticas válidas separadas por vírgulas. As políticas válidas são `delete` e `compact`. Para clusters habilitad os para armazenam ento hierárqui co, a política válida é somente. `delete`	Apache Kafka padrão

Propriedade	Descrição	Valor padrão
log.message.timestamp.after.max.ms	A diferença de data e hora permitida entre o timestamp da mensagem e o timestamp do corretor. O timestamp da mensagem pode ser posterior ou igual ao timestamp do broker, com a diferença máxima permitida determinada pelo valor definido nessa configuração. Se `log.message.timestamp.type=CreateTime`, a mensagem será rejeitada se a diferença nos timestamps exceder esse limite especificado. Essa configuração	86400000 (24 * 60 * 60 * 1000 ms, ou seja, 1 dia)

Propriedade	Descrição	Valor padrão
	é ignorada `selog.message.timestamp.type=LogAppendTime` .	

Propriedade	Descrição	Valor padrão
log.message.timestamp.before.max.ms	A diferença de data e hora permitida entre o timestamp do broker e o timestamp da mensagem. O timestamp da mensagem pode ser anterior ou igual ao timestamp do broker, com a diferença máxima permitida determinada pelo valor definido nessa configuração. Se `log.message.timestamp.type=CreateTime`, a mensagem será rejeitada se a diferença nos timestamps exceder esse limite especificado. Essa configuração	86400000 (24 * 60 * 60 * 1000 ms, ou seja, 1 dia)

Propriedade	Descrição	Valor padrão
	é ignorada `selog.message.timestamp.type=LogAppendTime` .	
log.message.timestamp.type	Especifica se o carimbo de data/hora na mensagem é o horário de criação da mensagem ou da adição no log. Os valores permitidos são `CreateTime` e `LogAppendTime` .	Apache Kafka padrão
log.retention.bytes	Tamanho máximo do log antes de ser excluído.	Apache Kafka padrão
log.retention.ms	Número de milissegundos para manter um arquivo de log antes de excluí-lo.	Apache Kafka padrão

Propriedade	Descrição	Valor padrão
max.connection.creation.rate	A taxa máxima de criação de conexão permitida na corretora a qualquer momento.	Apache Kafka padrão
conexões máximas	O número máximo de conexões permitidas no broker a qualquer momento. Esse limite é aplicado além de quaisquer limites por ip configurados usando `max.connections.per.ip` .	Apache Kafka padrão

Propriedade	Descrição	Valor padrão
max.connections.por ip	O número máximo de conexões permitidas de cada endereço IP. Isso pode ser definido como `0` se houver substituições configuradas usando a propriedade max.connections.per.ip.overrides. Novas conexões do endereço IP são descartadas se o limite for atingido.	Apache Kafka padrão

Propriedade	Descrição	Valor padrão
max.connections.per.ip.overrides	Uma lista separada por vírgula de nomes por IP ou host substitui o número máximo padrão de conexões. Um exemplo de valor é `hostName:100,127.0.0.1:200`	Apache Kafka padrão

Propriedade	Descrição	Valor padrão
message.max.bytes	O maior tamanho de lote de registros que o Kafka permite. Se você aumentar esse valor e houver consumidores anteriores à versão 0.10.2, também será necessário aumentar o tamanho de busca dos consumidores para que eles possam buscar lotes de registros desse tamanho. O formato de mensagem mais recente sempre agrupa as mensagens em lotes visando eficiência. As versões anteriores de formato de mensagem	Apache Kafka padrão

Propriedade	Descrição	Valor padrão
	não agrupam em lotes os registros não compactados, e, nesse caso, esse limite é aplicável somente a um único registro. Você pode definir esse valor por tópico com a `max.message.bytes` configuração do nível do tópico.	

Propriedade	Descrição	Valor padrão
producer.id.expiration.ms	O tempo em ms que um líder de partição de tópico esperará antes de expirar o produtor IDs. IDs O produtor não expirará enquanto uma transação associada a ele ainda estiver em andamento. Observe que o produtor IDs pode expirar mais cedo se a última gravação do ID do produtor for excluída devido às configurações de retenção do tópico. Definir esse valor igual ou superior a `delivery.timeout.ms` pode ajudar a evitar a expiração	Apache Kafka padrão

Propriedade	Descrição	Valor padrão
	durante novas tentativas e a proteger contra a duplicação de mensagens, mas o padrão deve ser razoável para a maioria dos casos de uso.	

Configurações em nível de tópico em Express Brokers

Você pode usar os comandos do Apache Kafka para definir ou modificar propriedades de configuração em nível de tópico para tópicos novos e existentes. Se você não puder fornecer nenhuma configuração em nível de tópico, o Amazon MSK usa o broker padrão. Assim como nas configurações em nível de corretor, o Amazon MSK protege algumas das propriedades de configuração em nível de tópico contra alterações. Os exemplos incluem fator de replicação `min.insync.replicas` e. `unclean.leader.election.enable` Se você tentar criar um tópico com um valor de fator de replicação diferente de`3`, o Amazon MSK criará o tópico com um fator de replicação de`3`, por padrão. Para obter mais informações sobre as propriedades de configuração no nível de tópico e exemplos sobre como defini-las, consulte Configurações no nível de tópico na documentação do Apache Kafka.

Propriedade	Descrição
cleanup.policy	Essa configuração designa a política de retenção a ser usada em segmentos de log. A política de “exclusão” (que é a padrão) descartará segmentos antigos quando seu tempo de retenção ou limite de tamanho for atingido. A política “compacta” permitirá a compactação de registros, que retém o valor mais recente de cada chave. Também é possível especificar as duas políticas

Propriedade	Descrição
	em uma lista separada por vírgulas (por exemplo, “excluir, compactar”). Nesse caso, os segmentos antigos serão descartados de acordo com a configuração de tamanho e tempo de retenção, enquanto os segmentos retidos serão compactados. A compactação em corretores Express é acionada depois que os dados em uma partição atingem 256 MB.
compression.type	Especifique o tipo de compactação final para um determinado tópico. Essa configura ção aceita os codecs de compressão padrão (`gzip`,, `snappylz4`,`zstd`). Além dissou`ncompressed` , aceita o que é equivalente a nenhuma compressão; e isso `producer` significa manter o codec de compressão original definido pelo produtor.
excluir.retention.ms	A quantidade de tempo necessário para reter marcadores de lápide excluídos para tópicos compactados de registros. Essa configuração também fornece um limite no tempo em que um consumidor deve concluir uma leitura se começar do deslocamento 0 para garantir que obtenha um instantâneo válido do estágio final. Caso contrário, as lápides excluídas podem ser coletadas antes que elas concluam a digitaliz ação. O valor padrão para essa configuração é 86400000 (24 * 60 * 60 * 1000 ms, ou seja, 1 dia), Apache Kafka Default

Propriedade	Descrição
max.message.bytes	O maior tamanho de lote de registros permitido pelo Kafka (após a compactação, se a compactação estiver ativada). Se isso aumentar e houver consumidores com mais de idade`0.10.2`, o tamanho da busca dos consumidores também deverá ser aumentado para que eles possam buscar lotes recordes desse tamanho. Na versão mais recente do formato de mensagem, os registros são sempre agrupados em lotes para obter eficiênci a. Nas versões anteriores do formato de mensagem, os registros não compactados não são agrupados em lotes, e, nesse caso, esse limite se aplica apenas a um único registro. Isso pode ser definido por tópico com o nível do tópico`max.message.bytes config` .
message.timestamp.after.max.ms	Essa configuração define a diferença de timestamp permitida entre o timestamp da mensagem e o timestamp do broker. O timestamp da mensagem pode ser posterior ou igual ao timestamp do broker, com a diferença máxima permitida determinada pelo valor definido nessa configuração. Se`message.t imestamp.type=CreateTime` , a mensagem será rejeitada se a diferença nos timestamps exceder esse limite especificado. Essa configuração é ignorada se`message.t imestamp.type=LogAppendTime` .

Propriedade	Descrição
message.timestamp.before.max.ms	Essa configuração define a diferença de timestamp permitida entre o timestamp do broker e o timestamp da mensagem. O timestamp da mensagem pode ser anterior ou igual ao timestamp do broker, com a diferença máxima permitida determinada pelo valor definido nessa configuração. Se `message.timestamp.type=CreateTime`, a mensagem será rejeitada se a diferença nos timestamps exceder esse limite especificado. Essa configuração é ignorada se `message.timestamp.type=LogAppendTime`.
message.timestamp.type	Defina se o carimbo de data/hora na mensagem é a hora de criação da mensagem ou a hora de acréscimo do registro. O valor deve ser `CreateTime` ou `LogAppendTime`
min.compaction.lag.ms	O tempo mínimo em que uma mensagem permanecerá descompactada no registro. Essa configuração é aplicável somente para registros que estão sendo compactados. O valor padrão para essa configuração é 0, Apache Kafka Default
max.compaction.lag.ms	O tempo máximo em que uma mensagem permanecerá inelegível para compactação no registro. Essa configuração é aplicável somente para registros que estão sendo compactados. Essa configuração seria limitada no intervalo de [7 dias, Long.Max]. O valor padrão para essa configuração é 9223372036854775807, Apache Kafka Default.

Propriedade	Descrição
retention.bytes	Essa configuração controla o tamanho máximo que uma partição (que consiste em segmentos de log) pode crescer antes de descartar mos segmentos de log antigos para liberar espaço se estivermos usando a política de retenção de “exclusão”. Por padrão, não há limite de tamanho, apenas um limite de tempo. Como esse limite é imposto no nível da partição, multiplique-o pelo número de partições para calcular a retenção do tópico em bytes. Além disso, `retention .bytes configuration` opera independe ntemente de `segment.bytes` configurações `segment.ms` e configurações. Além disso, ele aciona a rolagem de um novo segmento se ele `retention.bytes` estiver configurado para zero.
retention.ms	Essa configuração controla o tempo máximo que reteremos um registro antes de descartar mos segmentos de registro antigos para liberar espaço se estivermos usando a política de retenção de “exclusão”. Isso representa um SLA sobre a rapidez com que os consumido res devem ler seus dados. Se definido como `-1`, nenhum limite de tempo é aplicado. Além disso, a `retention.ms` configuração opera independentemente das `segment.ms` `segment.bytes` configurações. Além disso, aciona a rolagem de um novo segmento se a `retention.ms` condição for satisfeita.

Configurações somente para leitura do Express Brokers

O Amazon MSK define os valores para essas configurações e as protege contra alterações que possam afetar a disponibilidade do seu cluster. Esses valores podem mudar dependendo da versão do Apache Kafka em execução no cluster, portanto, lembre-se de verificar os valores do seu cluster específico. Aqui estão alguns exemplos.

Configurações somente para leitura do Express Brokers

Propriedade	Descrição	Valor expresso do corretor
broker.id	O ID do corretor desse servidor.	1,2,3...
corretor.rack	Prateleira do corretor. Isso será usado na atribuição de replicação com reconheci mento de rack para tolerância a falhas. Exemplos: `RACK1`, `us-east-1d`	ID AZ ou ID de sub-rede
default.replication.factor	Fatores de replicação padrão para todos os tópicos.	3
busque.max.bytes	O número máximo de bytes que retornaremos para uma solicitação de busca.	Apache Kafka padrão
group.max.size	O número máximo de consumidores que um único grupo de consumidores pode acomodar.	Apache Kafka padrão
inter.broker.listener.name	Nome do ouvinte usado para comunicação entre corretores.	REPLICATION_SECURE ou REPLICATION
inter.broker.protocol.version	Especifica qual versão do protocolo entre corretores é usada.	Apache Kafka padrão

Propriedade	Descrição	Valor expresso do corretor
Receptores	Lista de ouvintes - Lista separada por vírgulas dos nomes dos URIs ouvintes e os nomes dos ouvintes. Você pode definir o `advertised.listeners property` , mas não a `listeners` propriedade.	Gerado pelo MSK
log.message.format.version	Especifique a versão do formato da mensagem que o agente usará para anexar mensagens aos registros.	Apache Kafka padrão

Propriedade	Descrição	Valor expresso do corretor
min.insync.replicas	Quando um produtor define acks como `all` (ou `-1`), o valor em `min.insync.replicas` especifica o número mínimo de réplicas que devem reconhecer uma gravação para que a gravação seja considerada bem-sucedida. Se esse mínimo não puder ser atingido, o produtor cria uma exceção (`NotEnoughReplicas` ou `NotEnoughReplicasAfterAppend`). Você pode usar o valor dos pacotes de seu produtor para garantir maiores garantias de durabilidade. Ao definir pacotes como “todos”. Isso garante que o produtor gere uma exceção se a maioria das réplicas não receber uma gravação.	2
num.io.threads	Número de threads que o servidor usa para produzir solicitações, que podem incluir E/S de disco. (m7g.large, 8), (m7g.xlarge, 8), (m7g.2xlarge, 16), (m7g.4xlarge, 32), (m7g.8xlarge, 64), (m7g.12xlarge, 96), (m7g.16xlarge, 128)	Com base no tipo de instância. =Math.max (8, 2* v) CPUs

Propriedade	Descrição	Valor expresso do corretor
num.network.threads	Número de threads que o servidor usa para receber solicitações da rede e enviar respostas para a rede. (m7g.largo, 8), (m7g.xlargo, 8), (m7g.2xlargo, 8), (m7g.4xlargo, 16), (m7g.8xlargo, 32), (m7g.12xlargo, 48), (m7g.16xlargo, 64)	Com base no tipo de instância . =Math.max (8, v) CPUs
replica.fetch.response.max.bytes	O número máximo de bytes esperado para toda a resposta de busca. Os registros são buscados em lotes e, se o primeiro lote de registros na primeira partição não vazia da busca for maior que esse valor, o lote de registros ainda será retornado para garantir o progresso. Esse não é um máximo absoluto. As propriedades `message.max.bytes` (configuração do corretor) ou `max.message.bytes` (configuração do tópico) especificam o tamanho máximo do lote de registros que o corretor aceita.	Apache Kafka padrão

Propriedade	Descrição	Valor expresso do corretor
solicitação.timeout.ms	A configuração controla o tempo máximo que o cliente aguardará pela resposta de uma solicitação. Se a resposta não for recebida antes que o tempo limite termine, o cliente reenviará a solicitação, se necessário, ou falhará na solicitação se as novas tentativas forem esgotadas.	Apache Kafka padrão
transaction.state.log.min.isr	`min.insync.replicas` Configuração substituída para o tópico da transação.	2
transaction.state.log.replication.factor	O fator de replicação do tópico de transação.	Apache Kafka padrão
unclean.leader.election.enable	Permite que réplicas que não estão no conjunto ISR sirvam como líder como último recurso, mesmo que isso possa resultar em perda de dados.	FALSE

Operações de configuração do broker

As configurações do corretor Apache Kafka são estáticas ou dinâmicas. As configurações estáticas exigem a reinicialização do broker para que a configuração seja aplicada. As configurações dinâmicas não precisam ser reiniciadas pelo broker para que a configuração seja atualizada. Para obter mais informações sobre propriedades de configuração e modos de atualização, consulte Configuração do Apache Kafka.

Este tópico descreve como criar configurações personalizadas do MSK e como executar operações nelas. Para obter informações sobre como usar configurações do MSK para criar ou atualizar clusters, consulte the section called “Principais características e conceitos”.

Tópicos

Criar uma configuração

Este processo descreve como criar configurações personalizadas do Amazon MSK e como executar operações nelas.

1. Crie um arquivo para especificar as propriedades de configuração que você deseja definir e os valores que deseja atribuir a elas. Veja a seguir o conteúdo de um arquivo de configuração de exemplo.

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

log.roll.ms = 604800000
```

2. Execute o AWS CLI comando a seguir e *config-file-path* substitua pelo caminho para o arquivo em que você salvou sua configuração na etapa anterior.

> **Note**
>
> O nome que você escolher para sua configuração deve corresponder ao seguinte regex: "^[0-9A-Za-z][0-9A-Za-z-]{0,}$".

```
aws kafka create-configuration --name "ExampleConfigurationName" --description
 "Example configuration description." --kafka-versions "1.1.1" --server-properties
 fileb://config-file-path
```

Veja a seguir um exemplo de uma resposta bem-sucedida após a execução desse 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. O comando anterior retorna um nome do recurso da Amazon (ARN) para sua nova configuração. Salve esse ARN porque você precisará dele ao se referir a essa configuração em outros comandos. Se você perder o ARN da configuração, poderá listar todas as configurações da sua conta para encontrá-lo novamente.

Atualizar configuração

Este processo descreve como atualizar uma configuração personalizada do Amazon MSK.

1. Crie um arquivo para especificar as propriedades de configuração que você deseja atualizar e os valores que deseja atribuir a elas. Veja a seguir o conteúdo de um arquivo de configuração de exemplo.

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

min.insync.replicas = 2
```

2. Execute o AWS CLI comando a seguir e *config-file-path* substitua pelo caminho para o arquivo em que você salvou sua configuração na etapa anterior.

*configuration-arn*Substitua pelo ARN que você obteve ao criar a configuração. Se você não tiver salvado o ARN ao criar a configuração, poderá usar o comando `list-configurations` para listar todas as configurações em sua conta. A configuração que você deseja ver na lista aparecerá na resposta. O ARN da configuração também aparece nessa lista.

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

3. Veja a seguir um exemplo de uma resposta bem-sucedida após a execução desse 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
    }
}
```

Excluir configuração

O procedimento a seguir mostra como excluir uma configuração que não esteja anexada a um cluster. Não é possível excluir uma configuração anexada a um cluster.

1. Para executar esse exemplo, *configuration-arn* substitua pelo ARN obtido ao criar a configuração. Se você não tiver salvado o ARN ao criar a configuração, poderá usar o comando `list-configurations` para listar todas as configurações em sua conta. A configuração que você deseja ver na lista aparecerá na resposta. O ARN da configuração também aparece nessa lista.

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

2. Veja a seguir um exemplo de uma resposta bem-sucedida após a execução desse comando.

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

```
}
```

Obtenha metadados de configuração

O procedimento a seguir mostra como descrever uma configuração do Amazon MSK para obter metadados sobre ela.

1. O seguinte comando retornará metadados sobre a configuração. Para obter uma descrição detalhada da configuração, execute o `describe-configuration-revision`.

   Para executar esse exemplo, *`configuration-arn`* substitua pelo ARN obtido ao criar a configuração. Se você não tiver salvado o ARN ao criar a configuração, poderá usar o comando `list-configurations` para listar todas as configurações em sua conta. A configuração que você deseja ver na lista aparecerá na resposta. O ARN da configuração também aparece nessa lista.

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

2. Veja a seguir um exemplo de uma resposta bem-sucedida após a execução desse 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"
   }
   ```

Obtenha detalhes sobre a revisão da configuração

Este processo fornece uma descrição detalhada da revisão da configuração do Amazon MSK.

Se você usar o comando `describe-configuration` para descrever uma configuração do MSK, verá os metadados da configuração. Para obter uma descrição da configuração, use o comando `describe-configuration-revision`.

- Execute o comando a seguir e *`configuration-arn`* substitua pelo ARN obtido ao criar a configuração. Se você não tiver salvado o ARN ao criar a configuração, poderá usar o comando `list-configurations` para listar todas as configurações em sua conta. A configuração que você deseja ver na lista aparecerá na resposta. O ARN da configuração também aparece nessa lista.

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

  Veja a seguir um exemplo de uma resposta bem-sucedida após a execução desse 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
  }
  ```

  O valor de `ServerProperties` é codificado em base64. Se você usar um decodificador em base64 (por exemplo, https://www.base64decode.org/) para decodificá-lo manualmente, obterá o conteúdo do arquivo de configuração original usado para criar a configuração personalizada. Nesse caso, você obtém o seguinte:

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

  log.roll.ms = 604800000
  ```

Listar as configurações em sua conta para a região atual

Esse processo descreve como listar todas as configurações do Amazon MSK em sua conta para a região atual AWS .

- Execute o seguinte comando:

```
aws kafka list-configurations
```

Veja a seguir um exemplo de uma resposta bem-sucedida após a execução desse 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 das configurações do Amazon MSK

Uma configuração do Amazon MSK pode estar em um dos seguintes estados. Para realizar uma operação em uma configuração, a configuração deve estar no estado `ACTIVE` ou `DELETE_FAILED`:

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

Aplicação de patches

Aplicação de patches em clusters provisionados pelo MSK

Periodicamente, o Amazon MSK atualiza o software dos agentes do cluster. A manutenção inclui atualizações planejadas ou reparos não planejados. A manutenção planejada inclui atualizações do sistema operacional, atualizações de segurança e outras atualizações de software necessárias para manter a integridade, a segurança e o desempenho do seu cluster. Realizamos manutenção não planejada para resolver a degradação repentina da infraestrutura. Realizamos manutenção em corretores Standard e Express, mas as experiências são diferentes.

Correção de patches para corretores padrão

As atualizações dos agentes Standard não terão impacto nas gravações e leituras das aplicações se você seguir as práticas recomendadas.

O Amazon MSK usa atualizações contínuas de software para manter a alta disponibilidade dos clusters. Durante esse processo, os agentes são reiniciados um de cada vez e o Kafka transfere automaticamente a liderança para outro agente on-line. Os clientes Kafka têm mecanismos integrados para detectar automaticamente a mudança na liderança das partições e continuar gravando e lendo dados em um cluster do MSK. Siga Práticas recomendadas para clientes Apache Kafka para que seu cluster funcione sem problemas em todos os momentos, inclusive durante a aplicação de patches.

Depois que um agente fica offline, é normal ver erros transitórios de desconexão nos clientes. Você também observará por um breve período (até dois minutos, normalmente menos) alguns picos na latência de leitura e gravação de p99 (normalmente milissegundos altos, até aproximadamente dois segundos). Esses picos são esperados e são causados pela reconexão do cliente com um novo agente líder. Isso não afeta a produção ou o consumo e será resolvido após a reconexão. Para obter mais informações, consulte Agente offline e failover do cliente.

Você também observará um aumento na métrica`UnderReplicatedPartitions`, o que é esperado, pois as partições do agente que foi desligado não estão mais replicando dados. Isso não afeta as gravações e leituras das aplicações, pois as réplicas dessas partições hospedadas em outros agentes agora atendem às solicitações.

Após a atualização do software, quando o agente volta a ficar on-line, ele precisa “se atualizar” sobre as mensagens produzidas enquanto estava offline. Durante essa atualização, você também poderá observar um aumento no uso do throughput do volume e da CPU. Isso não deverá ter impacto nas gravações e leituras no cluster se você tiver recursos suficientes de CPU, memória, rede e volume nos agentes.

Correção de patches para corretores Express

Não há janelas de manutenção para corretores Express. O Amazon MSK atualiza automaticamente seu cluster de forma contínua e distribuída por tempo, o que significa que você pode esperar reinicializações ocasionais e únicas de agentes ao longo do mês. Isso garante que você não precise fazer planos ou acomodações em torno de janelas únicas de manutenção em todo o cluster. Como sempre, o tráfego permanecerá ininterrupto durante a reinicialização da corretora, pois a liderança mudará para outras corretoras que continuarão atendendo às solicitações.

Os corretores expressos vêm configurados com configurações de melhores práticas e grades de proteção que tornam seu cluster resiliente às mudanças de carga que podem ocorrer durante a manutenção. O Amazon MSK define cotas de taxa de transferência em seus agentes Express para mitigar o impacto da sobrecarga do seu cluster, o que pode causar problemas durante a reinicialização do agente. Essas melhorias eliminam a necessidade de notificações antecipadas, planejamento e janelas de manutenção quando você usa corretores Express.

Os corretores expressos sempre replicam seus dados de três maneiras para que seus clientes façam o failover automaticamente durante as reinicializações. Você não precisa se preocupar com a indisponibilidade dos tópicos devido ao fator de replicação definido como 1 ou 2. Além disso, o catch up de um corretor Express reiniciado é mais rápido do que em corretores Standard. A velocidade de aplicação de patches mais rápida nos corretores Express significa que haverá uma interrupção mínima no planejamento de qualquer atividade do plano de controle que você possa ter programado para seu cluster.

Como acontece com todos os aplicativos Apache Kafka, ainda há um contrato cliente-servidor compartilhado para clientes que se conectam aos corretores Express. Ainda é fundamental configurar seus clientes para lidar com a falha de liderança entre corretores. Siga a Práticas recomendadas para clientes Apache Kafka para uma operação tranquila do seu cluster em todos os

momentos, inclusive durante a aplicação de patches. Depois que um agente é reiniciado, é normal ver erros transitórios de desconexão nos clientes. Isso não afetará sua produção e consumo, pois os corretores seguidores assumirão a liderança da partição. Seus clientes do Apache Kafka farão o failover automaticamente e começarão a enviar solicitações aos novos agentes líderes.

Agente offline e failover do cliente

O Kafka permite um agente offline. Um único agente offline em um cluster íntegro e balanceado seguindo as práticas recomendadas não terá impacto nem causará falhas na produção ou no consumo. Isso ocorre porque outro agente assumirá a liderança da partição e a biblioteca do cliente Kafka fará o failover automaticamente e começará a enviar solicitações aos novos agentes líderes.

Contrato entre servidores e clientes

Isso resulta em um contrato compartilhado entre a biblioteca do cliente e o comportamento do servidor. O servidor deve atribuir com êxito um ou mais novos líderes e o cliente deve mudar de agente para enviar solicitações aos novos líderes em tempo hábil.

O Kafka usa exceções para controlar esse fluxo:

Um exemplo de procedimento

1. O agente A entra em um estado offline.
2. O cliente Kafka recebe uma exceção (normalmente uma desconexão de rede ou not_leader_for_partition).
3. Essas exceções acionam o cliente Kafka para que atualize os metadados para ter ciência dos líderes mais recentes.
4. O cliente Kafka retoma o envio de solicitações aos novos líderes de partições em outros agentes.

Esse processo normalmente leva menos de dois segundos com o cliente Java fornecido e as configurações padrão. Os erros do lado do cliente são redundantes e repetitivos, mas não são motivo de preocupação, conforme indicado pelo nível “WARN”.

Exemplo: exceção 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
```

Exemplo: exceção 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"
```

Os clientes Kafka resolverão automaticamente esses erros, normalmente em um segundo e, no máximo, três segundos. Isso se apresenta como produce/consume latência de p99 nas métricas do lado do cliente (normalmente altos milissegundos na década de 100). Um período maior do que isso normalmente indica um problema com a configuração do cliente ou com a carga do controlador do servidor. Consulte a seção de solução de problemas.

Um failover com êxito pode ser verificado ao conferir o aumento das métricas `BytesInPerSec` e `LeaderCount` e o aumento de outros agentes, o que prova que o tráfego e a liderança se moveram conforme o esperado. Você também observará um aumento na métrica `UnderReplicatedPartitions`, o que é esperado quando as réplicas estão offline com o agente de desligamento.

Solução de problemas

O fluxo acima pode ser interrompido pela quebra do contrato cliente-servidor. Os motivos mais comuns para o problema incluem:

- Configuração incorreta ou uso incorreto das bibliotecas do cliente Kafka.
- Comportamentos padrão inesperados e bugs com bibliotecas de clientes terceiros.
- Controlador sobrecarregado, resultando em uma atribuição mais lenta do líder de partição.
- Um novo controlador está sendo escolhido, resultando em uma atribuição mais lenta do líder de partição.

Para garantir um comportamento correto para lidar com failover de liderança, recomendamos:

- As práticas recomendadas do servidor devem ser seguidas para garantir que o agente controlador seja escalado adequadamente para evitar a demora na atribuição de liderança.
- As bibliotecas do cliente devem ter as novas tentativas habilitadas para garantir que o cliente trate o failover.
- As bibliotecas de cliente devem ter retry.backoff.ms configurado (padrão 100) para evitar tempestades. connection/request
- As bibliotecas do cliente devem definir request.timeout.ms e delivery.timeout.ms com valores em linha com o SLA das aplicações. Valores mais altos resultarão em um failover mais lento para determinados tipos de falha.
- As bibliotecas do cliente devem garantir que bootstrap.servers contenha pelo menos três agentes aleatórios para evitar um impacto na disponibilidade na descoberta inicial.
- Algumas bibliotecas de clientes têm um nível inferior ao de outras e esperam que o próprio desenvolvedor da aplicação implemente a lógica de repetição e o tratamento de exceções. Consulte a documentação específica da biblioteca do cliente para ver um exemplo de uso e certifique-se de que a reconnect/retry lógica correta seja seguida.
- Recomendamos monitorar a latência do lado do cliente quanto a produtos, à contagem com êxito de solicitações e à contagem de erros que não podem ser repetidos.
- Observamos que as bibliotecas mais antigas golang e ruby de terceiros permanecem redundantes durante todo o período offline do agente, apesar de as solicitações de produção e consumo não serem afetadas. Recomendamos que você sempre monitore as métricas em nível corporativo, além de solicitar métricas de êxito e erros, para determinar se há impacto real versus ruído nos logs.
- Os clientes não devem alertar sobre exceções transitórias de network/not_leader, pois elas são normais, não impactam e são esperadas como parte do protocolo kafka.
- Os clientes não devem se alarmar, UnderReplicatedPartitions pois são normais, não impactantes e esperados durante um único corretor off-line.

Registro em log do Amazon MSK

Você pode entregar registros do agente Apache Kafka para um ou mais dos seguintes tipos de destino: Amazon CloudWatch Logs, Amazon S3, Amazon Data Firehose. Você também pode registrar chamadas de API do Amazon MSK com AWS CloudTrail.

> **Note**
>
> Os registros do corretor não estão disponíveis nos corretores Express.

Logs do agente

Os logs de agente permitem solucionar problemas de aplicações do Apache Kafka e analisar as comunicações delas com o seu cluster do MSK. Você pode configurar seu cluster MSK novo ou existente para fornecer registros de agente em nível de informação a um ou mais dos seguintes tipos de recursos de destino: um grupo de CloudWatch registros, um bucket do S3, um stream de entrega do Firehose. Por meio do Firehose, você pode então entregar os dados de registro do seu stream de entrega para OpenSearch o Service. Você deve criar um recurso de destino antes de configurar seu cluster para entregar registros do agente a esse recurso. O Amazon MSK não cria esses recursos de destino para você se eles ainda não existirem. Para obter informações sobre esses três tipos de recursos de destino e como criá-los, consulte a seguinte documentação:

- CloudWatch Registros da Amazon
- Amazon S3
- Amazon Data Firehose

Permissões obrigatórias

Para configurar um destino para os logs de agente do Amazon MSK, a identidade do IAM que você usa para as ações do Amazon MSK deve ter as permissões descritas na política AWS política gerenciada: Amazon MSKFull Access.

Para transmitir logs de agente para um bucket do S3, também é necessário ter a permissão `s3:PutBucketPolicy`. Para obter informações sobre as políticas de bucket do S3, consulte Como adiciono uma política de bucket do S3? no Guia do usuário do Amazon S3. Para obter informações sobre as políticas do IAM em geral, consulte Gerenciamento de acesso no Guia do usuário do IAM.

Política de chave obrigatória do KMS para uso com buckets de SSE-KMS

Se você habilitou a criptografia do lado do servidor para seu bucket do S3 usando chaves AWS KMS gerenciadas (SSE-KMS) com uma chave gerenciada pelo cliente, adicione o seguinte à política de chaves da sua chave KMS para que o Amazon MSK possa gravar arquivos de agente no 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 os registros do broker usando o AWS Management Console

Se estiver criando um cluster, procure o cabeçalho Broker log delivery (Entrega de log de agente) na seção Monitoring (Monitoramento). É possível especificar os destinos aos quais deseja que o Amazon MSK entregue os logs de agente.

Para um cluster existente, escolha o cluster na lista de clusters e selecione a guia Propriedades. Role para baixo até a seção Entrega de logs e escolha o botão Editar. É possível especificar os destinos aos quais deseja que o Amazon MSK entregue os logs de agente.

Configure os registros do broker usando o AWS CLI

Quando você usar o comando `create-cluster` ou `update-monitoring`, poderá especificar o parâmetro `logging-info` opcionalmente e passar uma estrutura JSON para ele como o exemplo a seguir. Nesse JSON, todos os três tipos de destino são opcionais.

```
{
  "BrokerLogs": {
    "S3": {
      "Bucket": "amzn-s3-demo-bucket",
      "Prefix": "ExamplePrefix",
      "Enabled": true
    },
    "Firehose": {
      "DeliveryStream": "ExampleDeliveryStreamName",
      "Enabled": true
```

```
    },
    "CloudWatchLogs": {
      "Enabled": true,
      "LogGroup": "ExampleLogGroupName"
    }
  }
}
```

Configurar logs de agentes usando a API

Você pode especificar a `loggingInfo` estrutura opcional no JSON que você passa para as UpdateMonitoringoperações CreateClusterou.

> **Note**
>
> Por padrão, quando o registro em log do agente estiver habilitado, o Amazon MSK registrará os logs no nível de `INFO` nos destinos especificados. No entanto, os usuários do Apache Kafka 2.4.X e versões posteriores podem definir dinamicamente o nível de log do agente para qualquer um dos níveis de log log4j. Para obter informações sobre como definir dinamicamente o nível de log do agente, consulte KIP-412: estender a API Admin para oferecer suporte aos níveis dinâmicos de log do aplicativo. Se você definir dinamicamente o nível de log como `DEBUG` ou `TRACE`, recomendamos usar o Amazon S3 ou o Firehose como o destino de logs. Se você usar CloudWatch Logs como destino de log e ativar `DEBUG` ou `TRACE` nivelar dinamicamente o registro, o Amazon MSK poderá fornecer continuamente uma amostra de registros. Isso pode afetar significativamente o desempenho do agente e só deve ser usado quando o nível de log `INFO` não for suficientemente detalhado para determinar a causa raiz de um problema.

Registre chamadas de API com AWS CloudTrail

> **Note**
>
> AWS CloudTrail os registros estão disponíveis para o Amazon MSK somente quando você usaControle de acesso do IAM.

O Amazon MSK é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço no Amazon MSK. CloudTrail captura chamadas

de API como eventos. As chamadas capturadas incluem as chamadas do console do Amazon MSK e as chamadas de código para as operações da API do Amazon MSK. Ele também captura ações do Apache Kafka, como criar e alterar tópicos e grupos.

Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para o Amazon MSK. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita para a Amazon MSK ou a ação do Apache Kafka, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais CloudTrail, inclusive como configurá-lo e ativá-lo, consulte o Guia AWS CloudTrail do usuário.

Informações do Amazon MSK em CloudTrail

CloudTrail é ativado na sua conta da Amazon Web Services quando você cria a conta. Quando a atividade de evento suportada ocorre em um cluster MSK, essa atividade é registrada em um CloudTrail evento junto com outros eventos de AWS serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar os eventos recentes em sua conta da Amazon Web Services. Para obter mais informações, consulte Visualizar eventos com o histórico de eventos do CloudTrail.

Para obter um registro contínuo dos eventos na sua conta da Amazon Web Services, incluindo os eventos do Amazon MSK, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando uma trilha é criada no console, a mesma é aplicada a todas as regiões da . A trilha registra logs de eventos de todas as Regiões na AWS divisória e entrega os arquivos do log para o bucket Amazon S3 especificado. Além disso, você pode configurar outros serviços da Amazon para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para obter mais informações, consulte:

- Visão Geral para Criar uma Trilha
- CloudTrail Serviços e integrações compatíveis
- Configurando notificações do Amazon SNS para CloudTrail
- Recebendo arquivos de CloudTrail log de várias regiões e recebendo arquivos de CloudTrail log de várias contas

O Amazon MSK registra todas as operações do Amazon MSK como eventos em arquivos de CloudTrail log. Além disso, ele registra as seguintes ações do Apache Kafka.

- cluster de kafka: DescribeClusterDynamicConfiguration
- cluster de kafka: AlterClusterDynamicConfiguration
- cluster de kafka: CreateTopic
- cluster de kafka: DescribeTopicDynamicConfiguration
- cluster de kafka: AlterTopic
- cluster de kafka: AlterTopicDynamicConfiguration
- cluster de kafka: DeleteTopic

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar o seguinte:

- Se a solicitação foi feita com as credenciais do usuário root ou do usuário AWS Identity and Access Management (IAM).
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail .

Exemplo: entradas de arquivo de log do Amazon MSK

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas da API e das ações do Apache Kafka, portanto, eles não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra entradas de CloudTrail registro que demonstram as ações do `DeleteCluster` Amazon MSK `DescribeCluster` e do Amazon.

```
{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
```

```
        "principalId": "ABCDEF0123456789ABCDE",
        "arn": "arn:aws:iam::012345678901:user/Joe",
        "accountId": "012345678901",
        "accessKeyId": "AIDACKCEVSQ6C2EXAMPLE",
        "userName": "Joe"
      },
      "eventTime": "2018-12-12T02:29:24Z",
      "eventSource": "kafka.amazonaws.com",
      "eventName": "DescribeCluster",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "aws-cli/1.14.67 Python/3.6.0 Windows/10 botocore/1.9.20",
      "requestParameters": {
        "clusterArn": "arn%3Aaws%3Akafka%3Aus-east-1%3A012345678901%3Acluster
%2Fexamplecluster%2F01234567-abcd-0123-abcd-abcd0123efa-2"
      },
      "responseElements": null,
      "requestID": "bd83f636-fdb5-abcd-0123-157e2fbf2bde",
      "eventID": "60052aba-0123-4511-bcde-3e18dbd42aa4",
      "readOnly": true,
      "eventType": "AwsApiCall",
      "recipientAccountId": "012345678901"
    },
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "ABCDEF0123456789ABCDE",
        "arn": "arn:aws:iam::012345678901:user/Joe",
        "accountId": "012345678901",
        "accessKeyId": "AIDACKCEVSQ6C2EXAMPLE",
        "userName": "Joe"
      },
      "eventTime": "2018-12-12T02:29:40Z",
      "eventSource": "kafka.amazonaws.com",
      "eventName": "DeleteCluster",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "aws-cli/1.14.67 Python/3.6.0 Windows/10 botocore/1.9.20",
      "requestParameters": {
        "clusterArn": "arn%3Aaws%3Akafka%3Aus-east-1%3A012345678901%3Acluster
%2Fexamplecluster%2F01234567-abcd-0123-abcd-abcd0123efa-2"
      },
      "responseElements": {
```

```
        "clusterArn": "arn:aws:kafka:us-east-1:012345678901:cluster/
examplecluster/01234567-abcd-0123-abcd-abcd0123efa-2",
        "state": "DELETING"
      },
      "requestID": "c6bfb3f7-abcd-0123-afa5-293519897703",
      "eventID": "8a7f1fcf-0123-abcd-9bdb-1ebf0663a75c",
      "readOnly": false,
      "eventType": "AwsApiCall",
      "recipientAccountId": "012345678901"
    }
  ]
}
```

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `kafka-cluster:CreateTopic` ação.

```
{
  "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"
}
```

Gerenciamento de metadados

O Amazon MSK oferece suporte ao Apache ZooKeeper ou aos modos de gerenciamento de KRaft metadados.

A partir do Apache Kafka versão 3.7.x no Amazon MSK, você pode criar clusters que KRaft usam o modo em vez do modo. ZooKeeper KRaftclusters baseados em controladores no Kafka para gerenciar metadados.

Tópicos

- ZooKeeper modo
- KRaft modo

ZooKeeper modo

O Apache ZooKeeper é “um serviço centralizado para manter informações de configuração, nomear, fornecer sincronização distribuída e fornecer serviços de grupo. Todos esses tipos de serviços são usados de alguma forma por aplicações distribuídas”, incluindo o Apache Kafka.

Se seu cluster estiver usando o ZooKeeper modo, você pode usar as etapas abaixo para obter a string de ZooKeeper conexão do Apache. No entanto, recomendamos que você use `BootstrapServerString` para se conectar ao cluster e realizar operações administrativas, pois o sinalizador `--zookeeper` foi descontinuado no Kafka 2.5 e foi removido do Kafka 3.0.

Obtendo a string de ZooKeeper conexão do Apache usando o AWS Management Console

1. Abra o console do Amazon MSK em https://console.aws.amazon.com/msk/.
2. A tabela mostra todos os clusters da região atual nesta conta. Escolha o nome de um cluster para visualizar sua descrição.
3. Na página Resumo do cluster, escolha Exibir informações do cliente. Isso mostra os corretores de bootstrap, bem como a string de conexão do Apache ZooKeeper .

Obtendo a string de ZooKeeper conexão do Apache usando o AWS CLI

1. Se não souber o nome de recurso da Amazon (ARN) do cluster, você poderá encontrá-lo listando todos os clusters em sua conta. Para obter mais informações, consulte the section called “Listar clusters”.
2. Para obter a cadeia de ZooKeeper conexão do Apache, junto com outras informações sobre seu cluster, execute o comando a seguir, *ClusterArn* substituindo-o pelo ARN do seu cluster.

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

A saída desse comando `describe-cluster` é semelhante ao seguinte JSON de exemplo.

```
{
    "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"
    }
}
```

O JSON de exemplo anterior mostra a chave `ZookeeperConnectString` na saída do comando `describe-cluster`. Copie o valor correspondente a essa chave e salve-o para quando precisar criar um tópico no cluster.

> **Important**
>
> Seu cluster Amazon MSK deve estar no `ACTIVE` estado para que você possa obter a cadeia de ZooKeeper conexão Apache. Quando um cluster ainda está no estado `CREATING`, a saída do comando `describe-cluster` não inclui a `ZookeeperConnectString`. Se esse for o caso, aguarde alguns minutos e execute `describe-cluster` novamente após o cluster atingir o estado `ACTIVE`.

Obtendo a string de ZooKeeper conexão do Apache usando a API

Para obter a string de ZooKeeper conexão do Apache usando a API, consulte DescribeCluster.

KRaft modo

O Amazon MSK introduziu o suporte para KRaft (Apache Kafka Raft) na versão 3.7.x do Kafka. A comunidade Apache Kafka foi desenvolvida KRaft para substituir o Apache no gerenciamento de metadados nos clusters do Apache ZooKeeper Kafka. No KRaft modo, os metadados do cluster são propagados dentro de um grupo de controladores Kafka, que fazem parte do cluster Kafka, em vez de entre nós. ZooKeeper KRaftos controladores estão incluídos sem custo adicional para você e não exigem configuração ou gerenciamento adicionais de sua parte. Consulte KIP-500 para obter mais informações sobre. KRaft

Aqui estão alguns pontos a serem observados sobre o KRaft modo no MSK:

- KRaft o modo só está disponível para novos clusters. Não é possível alternar entre os modos de metadados depois que o cluster é criado.

- No console MSK, você pode criar um cluster baseado em Kraft escolhendo a versão 3.7.x do Kafka e marcando a caixa de seleção na janela de criação do cluster. KRaft
- Para criar um cluster no KRaft modo usando a API `CreateCluster`ou `CreateClusterV2`as operações do MSK, você deve usar `3.7.x.kraft` como versão. Use `3.7.x` como versão para criar um cluster no ZooKeeper modo.
- O número de partições por broker é o mesmo em clusters ZooKeeper baseados em KRaft e baseados. No entanto, KRaft permite que você hospede mais partições por cluster provisionando mais agentes em um cluster.
- Não são necessárias alterações de API para usar o KRaft modo no Amazon MSK. No entanto, se os clientes ainda usarem a string de conexão `--zookeeper` atualmente, você deverá atualizá-los para usar a string de conexão `--bootstrap-server` para se conectar ao cluster. Observe que o sinalizador `--zookeeper` foi descontinuado no Apache Kafka versão 2.5 e foi removido a partir do Kafka versão 3.0. Portanto, recomendamos que você use as versões recentes do cliente Apache Kafka e a string de conexão `--bootstrap-server` para todas as conexões com o cluster.
- ZooKeeper O modo continua disponível para todas as versões lançadas, nas quais o zookeeper também é suportado pelo Apache Kafka. Consulte Versões compatíveis do Apache Kafka para obter detalhes sobre o fim do suporte às versões do Apache Kafka e futuras atualizações.
- Você deve verificar se todas as ferramentas que você usa são capazes de usar o Kafka Admin APIs sem ZooKeeper conexões. Consulte Use o LinkedIn Cruise Control para Apache Kafka com o Amazon MSK para conferir as etapas atualizadas para conectar o cluster ao Cruise Control. O Cruise Control também tem instruções para executar o Cruise Control sem ZooKeeper.
- Você não precisa acessar os KRaft controladores do cluster diretamente para nenhuma ação administrativa. No entanto, se você estiver usando o monitoramento aberto para coletar métricas, também precisará dos endpoints de DNS dos controladores para coletar algumas métricas não relacionadas ao controlador sobre o cluster. Você pode obter esses endpoints de DNS no console do MSK ou usando a operação da ListNodesAPI. Consulte as etapas atualizadas Monitore um cluster provisionado pelo MSK com o Prometheus para configurar o monitoramento aberto para clusters KRaft baseados.
- Não há CloudWatch métricas adicionais que você precise monitorar para clusters de KRaft modos em vez de clusters ZooKeeper de modos. O MSK gerencia os KRaft controladores usados em seus clusters.
- Você pode continuar gerenciando ACLs usando clusters no KRaft modo usando a cadeia de `--bootstrap-server` conexão. Você não deve usar a cadeia de `--zookeeper` conexão para gerenciar ACLs. Consulte Apache Kafka ACLs.

- No KRaft modo, os metadados do seu cluster são armazenados em KRaft controladores dentro do Kafka e não em nós externos. ZooKeeper Portanto, você não precisa controlar o acesso aos nós do controlador separadamente, como você faz com ZooKeeper os nós.

Recursos do Amazon MSK

Dependendo do contexto, o termo recursos tem dois significados no Amazon MSK. No contexto de APIs um recurso, há uma estrutura na qual você pode invocar uma operação. Para obter uma lista desses recursos e das operações que você pode invocar neles, consulte Recursos na Referência de API do Amazon MSK. No contexto do the section called “Controle de acesso do IAM”, um recurso é uma entidade à qual você pode permitir ou proibir o acesso, conforme definido na seção the section called “Recursos da política de autorização”.

Versões do Apache Kafka

Ao criar um cluster do Amazon MSK, você especifica qual versão do Apache Kafka deseja que ele tenha. Também é possível atualizar a versão do Apache Kafka de um cluster existente. Os tópicos do capítulo ajudam você a entender as linhas do tempo do suporte à versão Kafka e as sugestões de práticas recomendadas.

Tópicos

- Versões compatíveis do Apache Kafka
- Suporte à versão do Amazon MSK

Versões compatíveis do Apache Kafka

O Amazon Managed Streaming for Apache Kafka (Amazon MSK) é compatível com as seguintes versões do Apache Kafka e do Amazon MSK. A comunidade do Apache Kafka fornece aproximadamente 12 meses de suporte para uma versão após sua data de lançamento. Para obter mais detalhes, consulte a política de EOL (fim da vida útil) do Apache Kafka.

Versões compatíveis do Apache Kafka

Versão do Apache Kafka	Data de lançamento do MSK	Data do fim do suporte
1.1.1	--	2024-06-05
2.1.0	--	2024-06-05

Versão do Apache Kafka	Data de lançamento do MSK	Data do fim do suporte
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	28/10/2022	2024-09-11
2,8.2-tiered	28/10/2022	2025-01-14
3.1.1	2022-06-22	2024-09-11
3.2.0	2022-06-22	2024-09-11
3.3.1	2022-10-26	2024-09-11
3.3.2	2023-03-02	2024-09-11
3.4.0	2023-05-04	2025-08-04

Versão do Apache Kafka	Data de lançamento do MSK	Data do fim do suporte
3.5.1	2023-09-26	23/10/2025-
3.6.0	2023-11-16	--
3.7.x	2024-05-29	--
3,8.x	2025-02-20	--
3.9.x	2025-04-21	--
4.0.x	2025-05-16	--

Para obter mais informações sobre a política de suporte à versão do Amazon MSK, consulte Política de suporte à versão do Amazon MSK.

Amazon MSK versão 4.0.x

O Amazon Managed Streaming for Apache Kafka (Amazon MSK) agora oferece suporte ao Apache Kafka versão 4.0. Essa versão traz os mais recentes avanços em gerenciamento e desempenho de clusters para o MSK Provisioned. O Kafka 4.0 introduz um novo protocolo de rebalanceamento do consumidor, agora disponível ao público em geral, que ajuda a garantir reequilíbrios de grupo mais suaves e rápidos. Além disso, o Kafka 4.0 exige que corretores e ferramentas usem o Java 17, fornecendo segurança e desempenho aprimorados, incluindo várias correções de bugs e melhorias e descontinuando o gerenciamento de metadados via Apache. ZooKeeper

Para obter mais detalhes e uma lista completa de melhorias e correções de erros, consulte as notas de lançamento do Apache Kafka para a versão 4.0.

Amazon MSK versão 3.9.x

O Amazon Managed Streaming for Apache Kafka (Amazon MSK) agora oferece suporte ao Apache Kafka versão 3.9. Essa versão permite que você retenha dados em camadas ao desativar o armazenamento em camadas no nível do tópico. Os aplicativos de consumo podem continuar lendo dados históricos do deslocamento remoto do início do registro (Rx) enquanto mantêm os deslocamentos contínuos do registro no armazenamento local e remoto.

Para obter mais detalhes e uma lista completa de melhorias e correções de erros, consulte as notas de lançamento do Apache Kafka para a versão 3.9.x.

Amazon MSK versão 3.8.x

O Amazon Managed Streaming for Apache Kafka (Amazon MSK) agora oferece suporte ao Apache Kafka versão 3.8. Agora você pode criar novos clusters usando a versão 3.8 com o KRAFT ou o ZooKeeper modo para gerenciamento de metadados ou atualizar seus clusters ZooKeeper baseados existentes para usar a versão 3.8. A versão 3.8 do Apache Kafka inclui várias correções de erros e novos recursos que melhoram o desempenho. Os principais novos recursos incluem suporte para configuração do nível de compressão. Isso permite que você otimize ainda mais seu desempenho ao usar tipos de compactação como lz4, zstd e gzip, permitindo que você altere o nível de compactação padrão.

Para obter mais detalhes e uma lista completa de melhorias e correções de erros, consulte as notas de lançamento do Apache Kafka para a versão 3.8.x.

Apache Kafka versão 3.7.x (com armazenamento em camadas pronto para produção)

O Apache Kafka versão 3.7.x no MSK inclui compatibilidade com o Apache Kafka versão 3.7.0. Você pode criar clusters ou atualizar clusters existentes para usar a nova versão 3.7.x. Com essa mudança no nome da versão, você não precisa mais adotar versões mais recentes de correção de patches, como a 3.7.1, quando forem lançadas pela comunidade do Apache Kafka. O Amazon MSK atualizará automaticamente a versão 3.7.x para ser compatível com as futuras versões de patch assim que elas estiverem disponíveis. Isso permite que você se beneficie da segurança e das correções de erros disponíveis nas versões de correção de patches sem acionar uma atualização de versão. Essas versões de correção de patches lançadas pelo Apache Kafka não quebram a compatibilidade de versões e você pode se beneficiar das novas versões de correção de patches sem se preocupar com erros de leitura ou gravação nas aplicações clientes. Certifique-se de que suas ferramentas de automação de infraestrutura, como CloudFormation, estejam atualizadas para considerar essa alteração na nomenclatura da versão.

O Amazon MSK agora oferece suporte ao KRaft modo (Apache Kafka Raft) no Apache Kafka versão 3.7.x. No Amazon MSK, assim como ZooKeeper nos nós, KRaft os controladores são incluídos sem custo adicional para você e não exigem configuração ou gerenciamento adicionais de sua parte. Agora você pode criar clusters em qualquer KRaft modo ou ZooKeeper modo no Apache Kafka versão 3.7.x. No modo KRaft, você pode adicionar até 60 agentes para hospedar mais partições por cluster, sem solicitar um aumento de limite, em comparação com a cota de 30 agentes em clusters baseados no Zookeeper. Para saber mais KRaft sobre o MSK, consulteKRaft modo.

A versão 3.7.x do Apache Kafka também inclui várias correções de erros e novos recursos que melhoram a performance. As principais melhorias incluem otimizações de descoberta de líderes para

clientes e opções de otimização de liberação de segmentos de logs. Para obter uma lista completa de melhorias e correções de erros, consulte as notas de lançamento do Apache Kafka para 3.7.0.

Apache Kafka versão 3.6.0 (com armazenamento em camadas pronto para produção)

Para obter informações sobre a versão 3.6.0 (com armazenamento em camadas pronto para produção) do Apache Kafka, consulte as notas de versão no site de downloads do Apache Kafka.

Para fins de estabilidade, o Amazon MSK continuará usando e gerenciando o Zookeeper para gerenciamento de quórum nesta versão.

Amazon MSK versão 3.5.1

O Amazon Managed Streaming for Apache Kafka (Amazon MSK) agora é compatível com a versão 3.5.1 do Apache Kafka para clusters novos e existentes. A versão 3.5.1 do Apache Kafka também inclui várias correções de erros e novos recursos que melhoram a performance. Os principais recursos incluem a introdução de uma nova atribuição de partições com reconhecimento de rack para consumidores. O Amazon MSK continuará a usar e gerenciar o Zookeeper para gerenciamento de quórum nesta versão. Para obter uma lista completa de melhorias e correções de erros, consulte as notas de lançamento do Apache Kafka para 3.5.1.

Para obter informações sobre a versão 3.5.1 do Apache Kafka, consulte as notas de versão no site de downloads do Apache Kafka.

Amazon MSK versão 3.4.0

O Amazon Managed Streaming for Apache Kafka (Amazon MSK) agora é compatível com a versão 3.4.0 do Apache Kafka para clusters novos e existentes. A versão 3.4.0 do Apache Kafka também inclui várias correções de erros e novos recursos que melhoram a performance. Os principais recursos incluem uma correção para melhorar a estabilidade da busca na réplica mais próxima. O Amazon MSK continuará a usar e gerenciar o Zookeeper para gerenciamento de quórum nesta versão. Para obter uma lista completa de melhorias e correções de erros, consulte as notas de lançamento do Apache Kafka para 3.4.0.

Para obter informações sobre a versão 3.4.0 do Apache Kafka, consulte as notas de versão no site de downloads do Apache Kafka.

Amazon MSK versão 3.3.2

O Amazon Managed Streaming for Apache Kafka (Amazon MSK) agora é compatível com a versão 3.3.2 do Apache Kafka para clusters novos e existentes. A versão 3.3.2 do Apache Kafka também

inclui várias correções de erros e novos recursos que melhoram a performance. Os principais recursos incluem uma correção para melhorar a estabilidade da busca na réplica mais próxima. O Amazon MSK continuará a usar e gerenciar o Zookeeper para gerenciamento de quórum nesta versão. Para obter uma lista completa de melhorias e correções de erros, consulte as notas de lançamento do Apache Kafka para 3.3.2.

Para obter informações sobre a versão 3.3.2 do Apache Kafka, consulte as notas de versão no site de downloads do Apache Kafka.

Amazon MSK versão 3.3.1

O Amazon Managed Streaming for Apache Kafka (Amazon MSK) agora é compatível com a versão 3.3.1 do Apache Kafka para clusters novos e existentes. A versão 3.3.1 do Apache Kafka também inclui várias correções de erros e novos recursos que melhoram a performance. Alguns dos principais recursos incluem aprimoramentos nas métricas e no particionador. Para fins de estabilidade, o Amazon MSK continuará usando e gerenciando o Zookeeper para gerenciamento de quórum nesta versão. Para obter uma lista completa de melhorias e correções de erros, consulte as notas de lançamento do Apache Kafka para 3.3.1.

Para obter informações sobre a versão 3.3.1 do Apache Kafka, consulte as notas de versão no site de downloads do Apache Kafka.

Amazon MSK versão 3.1.1

O Amazon Managed Streaming for Apache Kafka (Amazon MSK) agora é compatível com a versões 3.1.1 e 3.2.0 do Apache Kafka para clusters novos e existentes. As versões 3.1.1 e 3.2.0 do Apache Kafka também incluem várias correções de erros e novos recursos que melhoram a performance. Alguns dos principais recursos incluem aprimoramentos nas métricas e no uso do tópico. IDs O MSK continuará a usar e gerenciar o Zookeeper para gerenciamento de quórum nesta versão para fins de estabilidade. Para obter uma lista completa de melhorias e correções de erros, consulte as notas de lançamento do Apache Kafka para 3.1.1 e 3.2.0.

Para obter informações sobre as versões 3.1.1 e 3.2.0 do Apache Kafka, consulte as notas de lançamento da versão 3.2.0 e as notas de lançamento da versão 3.1.1 no site de downloads do Apache Kafka.

Armazenamento em camadas do Amazon MSK versão 2.8.2.tiered

Essa versão é uma versão exclusiva do Amazon MSK do Apache Kafka versão 2.8.2, sendo compatível com clientes Apache Kafka de código aberto.

A versão 2.8.2.tiered contém a funcionalidade de armazenamento em camadas que é compatível com a APIs introduzida no KIP-405 para Apache Kafka. Para obter mais informações sobre o recurso de armazenamento em camadas do Amazon MSK, consulte Armazenamento hierárquico para corretores padrão.

Apache Kafka versão 2.5.1

A versão 2.5.1 do Apache Kafka inclui várias correções de erros e novos recursos, incluindo criptografia em trânsito para clientes Apache e de administração. ZooKeeper O Amazon MSK fornece ZooKeeper endpoints TLS, que você pode consultar com a operação. DescribeCluster

A saída da DescribeClusteroperação inclui o `ZookeeperConnectStringTls` nó, que lista os endpoints do TLS zookeeper.

O exemplo a seguir mostra o nó `ZookeeperConnectStringTls` da resposta para a operação `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 obter informações sobre o uso da criptografia TLS com o zookeeper, consulte Usando a segurança TLS com o Apache ZooKeeper.

Para obter mais informações sobre a versão 2.5.1 do Apache Kafka, consulte as notas de versão no site de downloads do Apache Kafka.

Correção de bugs do Amazon MSK versão 2.4.1.1

Essa versão é uma versão de correção de bugs do Apache Kafka versão 2.4.1 exclusiva do Amazon MSK. Essa versão de correção de bugs contém uma correção para o KAFKA-9752, um problema raro que faz com que grupos de consumidores façam o rebalanceamento contínuo e permaneçam no estado `PreparingRebalance`. Esse problema afeta clusters que executam as versões 2.3.1 e 2.4.1. Essa versão contém uma correção produzida pela comunidade que está disponível na versão 2.5.0 do Apache Kafka.

> **Note**
>
> Os clusters do Amazon MSK que executam a versão 2.4.1.1 são compatíveis com qualquer cliente Apache Kafka compatível com o Apache Kafka versão 2.4.1.

Recomendamos que você use a correção de bugs do MSK versão 2.4.1.1 para novos clusters do Amazon MSK se preferir usar o Apache Kafka 2.4.1. É possível atualizar os clusters existentes que executam o Apache Kafka versão 2.4.1 para essa versão a fim de incorporar essa correção. Para obter informações sobre como atualizar um cluster existente, consulte Atualize a versão do Apache Kafka.

Para contornar esse problema sem atualizar o cluster para a versão 2.4.1.1, consulte a seção Grupo de consumidores preso no estado `PreparingRebalance` do guia Solução de problemas para o cluster do Amazon MSK.

Apache Kafka versão 2.4.1 (use 2.4.1.1 alternativamente)

> **Note**
>
> Você não pode mais criar um cluster do MSK com o Apache Kafka versão 2.4.1. Em vez disso, você pode usar a versão Correção de bugs do Amazon MSK versão 2.4.1.1 com clientes compatíveis com o Apache Kafka versão 2.4.1. E se você já tiver um cluster do MSK com o Apache Kafka versão 2.4.1, recomendamos que você o atualize para usar o Apache Kafka versão 2.4.1.1.

O KIP-392 é uma das principais propostas de melhoria do Kafka incluídas na versão 2.4.1 do Apache Kafka. Essa melhoria permite que os consumidores busquem a partir da réplica mais próxima. Para usar esse recurso, defina `client.rack` nas propriedades do consumidor como o ID da zona de disponibilidade do consumidor. Um exemplo de ID AZ é `use1-az1`. O Amazon MSK define `broker.rack` as zonas IDs de disponibilidade dos corretores. Também é necessário definir a propriedade de configuração `replica.selector.class` como `org.apache.kafka.common.replica.RackAwareReplicaSelector`, que é uma implementação de reconhecimento de rack fornecida pelo Apache Kafka.

Quando você usa esta versão do Apache Kafka, as métricas no nível de monitoramento `PER_TOPIC_PER_BROKER` aparecem somente após os valores se tornarem diferentes de zero pela

primeira vez. Para obter mais informações sobre isso, consulte the section called “Monitoramento no nível PER_TOPIC_PER_BROKER”.

Para obter informações sobre como encontrar a Zona de Disponibilidade IDs, consulte AZ IDs for Your Resource no guia AWS Resource Access Manager do usuário.

Para obter informações sobre como definir propriedades de configuração, consulte the section called “Configuração do corretor”.

Para obter mais informações sobre o KIP-392, consulte Permitir que os consumidores busquem a partir da réplica mais próxima nas páginas do Confluence.

Para obter mais informações sobre a versão 2.4.1 do Apache Kafka, consulte as notas de release no site de downloads do Apache Kafka.

Suporte à versão do Amazon MSK

Este tópico descreve a Política de suporte à versão do Amazon MSK e o procedimento para Atualize a versão do Apache Kafka. Se você estiver atualizando sua versão do Kafka, siga as práticas recomendadas descritas em Práticas recomendadas para upgrades de versão.

Tópicos

- Política de suporte à versão do Amazon MSK
- Atualize a versão do Apache Kafka
- Práticas recomendadas para upgrades de versão

Política de suporte à versão do Amazon MSK

Esta seção descreve a política de suporte para as versões do Kafka compatíveis com o Amazon MSK.

- Todas as versões do Kafka são compatíveis até atingirem a data do fim do suporte. Para obter detalhes sobre as datas de fim do suporte, consulte Versões compatíveis do Apache Kafka. Atualize o cluster do MSK para a versão recomendada do Kafka ou superior antes da data do fim do suporte. Para obter detalhes sobre como atualizar sua versão do Apache Kafka, consulte. Atualize a versão do Apache Kafka Um cluster usando uma versão do Kafka após a data do fim do suporte é atualizado automaticamente para a versão recomendada do Kafka. As atualizações automáticas podem ocorrer a qualquer momento após a data de término do suporte. Você não receberá nenhuma notificação antes da atualização.

- O MSK descontinuará gradualmente o suporte para clusters recém-criados que usam versões do Kafka com datas de fim de suporte publicadas.

Atualize a versão do Apache Kafka

Você pode atualizar um cluster MSK existente para uma versão mais recente do Apache Kafka.

> **Note**
>
> - Você não pode atualizar um cluster MSK existente de uma versão ZooKeeper baseada no Apache Kafka para uma versão mais recente que use ou exija o modo. KRaft Em vez disso, para atualizar seu cluster, crie um novo cluster MSK com uma versão KRaft compatível com o Kafka e migre seus dados e cargas de trabalho do cluster antigo.
> - O Amazon MSK atualiza somente o software do servidor. Isso não atualiza seus clientes.
> - Você não pode fazer o downgrade de um cluster MSK existente para uma versão mais antiga do Apache Kafka.

Ao atualizar a versão Apache Kafka de um cluster MSK, verifique também o software do lado do cliente para garantir que sua versão permita que você use os recursos da nova versão do Apache Kafka do cluster.

Para obter informações sobre como tornar um cluster altamente disponível durante uma atualização, consultethe section called “Criar clusters altamente disponíveis”.

Atualize a versão do Apache Kafka usando o AWS Management Console

1. Abra o console do Amazon MSK em https://console.aws.amazon.com/msk/.
2. Na barra de navegação, escolha a região em que você criou o cluster MSK.
3. Escolha o cluster MSK que você deseja atualizar.
4. Na guia Propriedades, escolha Atualizar na seção Versão do Apache Kafka.
5. Na seção de versão do Apache Kafka, faça o seguinte:

    a. Na lista suspensa Escolha a versão do Apache Kafka, escolha a versão para a qual você deseja atualizar. Para este exemplo, selecione **`3.9.x`**.

b. (Opcional) Escolha Compatibilidade de versão para ver a compatibilidade entre a versão atual do cluster e a versão para a qual você deseja fazer o upgrade. Em seguida, escolha Escolher para continuar ou escolha Cancelar.

c. Escolha a caixa de seleção Atualizar configuração do cluster para aplicar automaticamente uma nova revisão de configuração do Kafka que seja compatível com a versão atualizada. Isso garante a compatibilidade e permite novos recursos ou melhorias na versão atualizada. No entanto, ignore-a se quiser manter suas configurações personalizadas existentes.

d. Escolha Atualizar.

Atualize a versão do Apache Kafka usando o AWS CLI

1. Execute o comando a seguir, substituindo *ClusterArn* pelo nome do recurso da Amazon (ARN) que você obteve quando criou o cluster. Se você não tiver o ARN do cluster, poderá encontrá-lo listando todos os clusters. Para obter mais informações, consulte the section called "Listar clusters".

```
aws kafka get-compatible-kafka-versions --cluster-arn ClusterArn
```

A saída desse comando inclui uma lista das versões do Apache Kafka para as quais você pode atualizar o cluster. Ela se parece com o exemplo a seguir.

```
{
    "CompatibleKafkaVersions": [
        {
            "SourceVersion": "2.2.1",
            "TargetVersions": [
                "2.3.1",
                "2.4.1",
                "2.4.1.1",
                "2.5.1"
            ]
        }
    ]
}
```

2. Execute o comando a seguir, substituindo *ClusterArn* pelo nome do recurso da Amazon (ARN) que você obteve quando criou o cluster. Se você não tiver o ARN do cluster, poderá

encontrá-lo listando todos os clusters. Para obter mais informações, consulte the section called "Listar clusters".

Substitua *Current-Cluster-Version* pela versão atual do cluster. Pois *TargetVersion* você pode especificar qualquer uma das versões de destino a partir da saída do comando anterior.

> **Important**
>
> As versões de cluster não são inteiros simples. Para encontrar a versão atual do cluster, use a DescribeClusteroperação ou o comando AWS CLI describe-cluster. Uma versão de exemplo é `KTVPDKIKX0DER`.

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

A saída do comando anterior é semelhante ao JSON a seguir.

```
{

    "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 obter o resultado da `update-cluster-kafka-version` operação, execute o comando a seguir, *ClusterOperationArn* substituindo-o pelo ARN obtido na saída do `update-cluster-kafka-version` comando.

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

A saída desse comando `describe-cluster-operation` é semelhante ao seguinte JSON de exemplo.

```
{
    "ClusterOperationInfo": {
```

```
        "ClientRequestId": "62cd41d2-1206-4ebf-85a8-dbb2ba0fe259",
        "ClusterArn": "arn:aws:kafka:us-east-1:012345678012:cluster/
exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2",
        "CreationTime": "2021-03-11T20:34:59.648000+00:00",
        "OperationArn": "arn:aws:kafka:us-east-1:012345678012:cluster-
operation/exampleClusterName/abcdefab-1234-abcd-5678-cdef0123ab01-2/0123abcd-
abcd-4f7f-1234-9876543210ef",
        "OperationState": "UPDATE_IN_PROGRESS",
        "OperationSteps": [
            {
                "StepInfo": {
                    "StepStatus": "IN_PROGRESS"
                },
                "StepName": "INITIALIZE_UPDATE"
            },
            {
                "StepInfo": {
                    "StepStatus": "PENDING"
                },
                "StepName": "UPDATE_APACHE_KAFKA_BINARIES"
            },
            {
                "StepInfo": {
                    "StepStatus": "PENDING"
                },
                "StepName": "FINALIZE_UPDATE"
            }
        ],
        "OperationType": "UPDATE_CLUSTER_KAFKA_VERSION",
        "SourceClusterInfo": {
            "KafkaVersion": "2.4.1"
        },
        "TargetClusterInfo": {
            "KafkaVersion": "2.6.1"
        }
    }
}
```

Se `OperationState` tiver o valor `UPDATE_IN_PROGRESS`, aguarde um pouco e execute o comando `describe-cluster-operation` novamente. Quando a operação for concluída, o valor de `OperationState` será transformado em `UPDATE_COMPLETE`. Como o tempo

necessário para que o Amazon MSK conclua a operação varia, talvez seja necessário verificar repetidamente até que a operação seja concluída.

Atualize a versão do Apache Kafka usando a API

1. Invoque a GetCompatibleKafkaVersionsoperação para obter uma lista das versões do Apache Kafka para as quais você pode atualizar o cluster.
2. Invoque a UpdateClusterKafkaVersionoperação para atualizar o cluster para uma das versões compatíveis do Apache Kafka.

Práticas recomendadas para upgrades de versão

Para garantir a continuidade do cliente durante a atualização contínua que é realizada como parte do processo de atualização da versão do Kafka, revise a configuração dos clientes e os tópicos do Apache Kafka da seguinte forma:

- Defina o fator de replicação (RF) do tópico para um valor mínimo de 2 para clusters de duas AZs e um valor mínimo de 3 para clusters de três AZs. Um valor de RF de 2 pode levar a partições offline durante a aplicação de patches.
- Defina o mínimo de réplicas sincronizadas (miniSR) para um valor máximo de 1 a menos do que seu Fator de Replicação (RF), que é. `miniISR = (RF) - 1` Isso garante que o conjunto de réplicas de partições possa tolerar que uma réplica fique off-line ou sub-replicada.
- Configure os clientes para usar várias strings de conexão de agentes. Ter vários corretores na cadeia de conexão de um cliente permite o failover se um corretor específico que dá suporte ao cliente I/O começar a ser corrigido. Para obter informações sobre como obter uma string de conexão com vários agentes, consulte Obter os agentes de bootstrap para um cluster do Amazon MSK.
- Recomendamos que você atualize os clientes de conexão para a versão recomendada ou superior para se beneficiar dos recursos disponíveis na nova versão. As atualizações do cliente não estão sujeitas às datas de fim da vida útil (EOL) da versão Kafka do cluster do MSK e não precisam ser concluídas até a data de EOL. O Apache Kafka fornece uma política bidirecional de compatibilidade de clientes que permite que clientes mais antigos trabalhem com clusters mais novos, e vice-versa.
- Os clientes Kafka que usam as versões 3.x.x provavelmente virão com os seguintes padrões: `acks=all` e `enable.idempotence=true`. `acks=all` é diferente do padrão anterior de `acks=1` e fornece durabilidade extra ao garantir que todas as réplicas sincronizadas reconheçam

a solicitação de produção. Da mesma forma, o padrão para `enable.idempotence` era anteriormente `false`. A alteração para `enable.idempotence=true` como o padrão reduz a probabilidade de mensagens duplicadas. Essas alterações são consideradas configurações de práticas recomendadas e podem introduzir uma pequena quantidade de latência adicional que está dentro dos parâmetros normais de performance.

- Use a versão recomendada do Kafka ao criar clusters do MSK. Usar a versão recomendada do Kafka permite que você se beneficie dos recursos mais recentes do Kafka e do MSK.

Solução de problemas para o cluster do Amazon MSK

As informações a seguir podem ajudar você a solucionar problemas que possam surgir com seu cluster do Amazon MSK. Você também pode publicar seu problema no AWS re:Post. Para a solução de problemas do Replicador do Amazon MSK, consulte Solucionar problemas do Replicador do MSK.

Tópicos

- Não é possível acessar o cluster de dentro AWS: problemas de rede
- Falha na autenticação: muitas conexões
- Falha na autenticação: sessão muito curta
- MSK com tecnologia sem servidor: falha na criação do cluster
- Não é possível atualizar KafkaVersionsList na configuração do MSK

A substituição do volume causa a saturação do disco devido à sobrecarga de replicação

Durante uma falha de hardware de volume não planejada, o Amazon MSK pode substituir o volume por uma nova instância. O Kafka preenche novamente o novo volume replicando partições de outros agentes no cluster. Depois que as partições são replicadas e recuperadas, elas se qualificam para associação de liderança e réplica em sincronia (ISR).

Problema

Em um agente se recuperando da substituição de volume, algumas partições de tamanhos variados podem voltar a ficar on-line antes de outras. Isso pode ser problemático, pois essas partições podem estar fornecendo tráfego do mesmo agente que ainda está recuperando (replicando) outras partições. Às vezes, esse tráfego de replicação pode saturar os limites de throughput do volume subjacente, que são de 250 MiB por segundo no caso padrão. Quando essa saturação ocorre, todas as partições que já estão atualizadas serão afetadas, resultando em latência em todo o cluster para qualquer agente que compartilhe a ISR com essas partições atualizadas (não apenas partições líderes devido a acks remotos `acks=all`). Esse problema é mais comum em clusters maiores que têm um número maior de partições que variam em tamanho.

Recomendação

- Para melhorar a I/O postura de replicação, certifique-se de que as configurações de thread de melhores práticas estejam em vigor.
- Para reduzir a probabilidade de saturação do volume subjacente, habilite o armazenamento provisionado com um throughput mais alto. Um valor mínimo de taxa de transferência de 500 MiB/s é recomendado para casos de replicação de alta taxa de transferência, mas o valor real necessário variará de acordo com a taxa de transferência e o caso de uso. Provisione a taxa de transferência de armazenamento para corretores padrão em um cluster Amazon MSK.
- Para minimizar a pressão de replicação, reduza `num.replica.fetchers` para o valor padrão de 2.

Grupo de consumidores preso no estado `PreparingRebalance`

Se um ou mais de seus grupos de consumidores estiverem presos em um estado perpétuo de rebalanceamento, a causa disso pode ser o problema KAFKA-9752 do Apache Kafka, que afeta as versões 2.3.1 e 2.4.1 do Apache Kafka.

Para solucionar esse problema, recomendamos que você atualize seu cluster para a versão Correção de bugs do Amazon MSK versão 2.4.1.1, que contém uma correção para esse problema. Para obter informações sobre a atualização de um cluster existente para a versão 2.4.1.1 de correção de bugs do Amazon MSK, consulte Atualize a versão do Apache Kafka.

As soluções alternativas para resolver esse problema sem atualizar o cluster para a versão 2.4.1.1 de correção de bugs do Amazon MSK são definir os clientes do Kafka para usar Protocolo de associação estática ou Identificar e reiniciar o nó do agente de coordenação do grupo de consumidores que está preso.

Implementação de protocolo de associação estática

Para implementar o protocolo de associação estática em seus clientes, faça o seguinte:

1. Defina a propriedade `group.instance.id` da sua configuração Consumidores do Kafka como uma string estática que identifica o consumidor no grupo.
2. Certifique-se de que outras instâncias da configuração sejam atualizadas para usar a string estática.
3. Implante as mudanças em seus consumidores do Kafka.

O uso do Protocolo de associação estática é mais eficaz se o tempo limite da sessão na configuração do cliente for definido para uma duração que permita ao consumidor se recuperar sem acionar prematuramente um rebalanceamento do grupo de consumidores. Por exemplo, se sua aplicação consumidora conseguir tolerar 5 minutos de indisponibilidade, um valor razoável para o tempo limite da sessão seria 4 minutos em vez do valor padrão de 10 segundos.

> **Note**
>
> O uso do protocolo de associação estática simplesmente reduz a probabilidade de se deparar com esse problema. Você ainda poderá se deparar com esse problema mesmo ao usar o protocolo de associação estática.

Como reinicializar o nó do agente de coordenação

Para reinicializar o nó agente de coordenação, faça o seguinte:

1. Identifique o coordenador do grupo usando o comando `kafka-consumer-groups.sh`.
2. Reinicie o coordenador do grupo de consumidores bloqueados usando a ação RebootBrokerda API.

Erro ao entregar os registros do corretor para o Amazon CloudWatch Logs

Ao tentar configurar seu cluster para enviar registros do agente para a Amazon CloudWatch Logs, você pode obter uma das duas exceções.

Se você receber uma exceção `InvalidInput.LengthOfCloudWatchResourcePolicyLimitExceeded`, tente novamente, mas use grupos de log que começam com `/aws/vendedlogs/`. Para obter mais informações, consulte Habilitar o registro em log de determinados serviços da Amazon Web Services.

Se você receber uma `InvalidInput.NumberOfCloudWatchResourcePoliciesLimitExceeded` exceção, escolha uma política existente do Amazon CloudWatch Logs em sua conta e acrescente o seguinte JSON a ela.

```
{"Sid":"AWSLogDeliveryWrite","Effect":"Allow","Principal":
{"Service":"delivery.logs.amazonaws.com"},"Action":
["logs:CreateLogStream","logs:PutLogEvents"],"Resource":["*"]}
```

Se você tentar anexar o JSON acima a uma política existente, mas receber um erro informando que você atingiu o tamanho máximo da política escolhida, tente anexar o JSON a outra de suas políticas do Amazon Logs. CloudWatch Depois de acrescentar o JSON a uma política existente, tente novamente configurar a entrega de registros do corretor para o Amazon Logs. CloudWatch

Nenhum grupo de segurança padrão

Se você tentar criar um cluster e obter um erro indicando que não há grupo de segurança padrão, talvez esteja usando uma VPC que foi compartilhada com você. Peça para o administrador conceder permissão para descrever os grupos de segurança nesta VPC e tente novamente. Para ver um exemplo de uma política que permite essa ação, consulte Amazon EC2: Permite gerenciar grupos de EC2 segurança associados a uma VPC específica, programaticamente e no console.

O cluster parece estar preso no estado CRIANDO

Às vezes a criação do cluster pode levar até 30 minutos. Aguarde 30 minutos e verifique o estado do cluster novamente.

O estado do cluster é alterado de CRIANDO para COM FALHA

Tente criar o cluster novamente.

O estado do cluster está ATIVO, mas os produtores não conseguem enviar dados ou os consumidores não conseguem receber dados

- Se a criação do cluster tiver êxito (o estado do cluster será `ACTIVE`), mas não será possível enviar nem receber dados. Certifique-se de que os aplicativos produtor e consumidor tenham acesso ao cluster. Para obter mais informações, consulte as diretrizes no the section called “Criar uma máquina cliente”.

- Caso os produtores e os consumidores tenham acesso ao cluster, mas ainda assim enfrentem problemas ao gerar e consumir dados, a causa pode ser KAFKA-7697, que afeta o Apache Kafka versão 2.1.0 e pode levar a um deadlock em um ou mais agentes. Considere migrar para o Apache Kafka 2.2.1, que não é afetado por este bug. Para obter informações sobre como migrar, consulte the section called “Migrar para um cluster do Amazon MSK”.

AWS CLI não reconhece o Amazon MSK

Se você o tiver AWS CLI instalado, mas ele não reconhecer os comandos do Amazon MSK, atualize-o AWS CLI para a versão mais recente. Para obter instruções detalhadas sobre como atualizar o AWS CLI, consulte Instalando AWS Command Line Interface o. Para obter informações sobre como usar os comandos AWS CLI para executar o Amazon MSK, consultethe section called “Principais características e conceitos”.

As partições ficam offline ou as réplicas estão fora de sincronia

Estes podem ser sintomas de pouco espaço em disco. Consulte the section called “O espaço em disco está acabando”.

O espaço em disco está acabando

Consulte as melhores práticas para gerenciar o espaço em disco: the section called “Monitorar o espaço em disco” e the section called “Ajustar os parâmetros de retenção de dados”.

A memória está baixa

Caso a métrica `MemoryUsed` esteja alta ou a `MemoryFree` esteja baixa, isso não significa que existe um problema. O Apache Kafka foi desenvolvido para usar o máximo de memória possível, que é gerenciada de forma ideal.

O produtor recebe NotLeaderForPartitionException

Geralmente, isto é um erro transitório. Defina o parâmetro de configuração de `retries` do produtor com um valor mais alto que o atual.

Número de partições com replicação insuficiente (URP) maior que zero

A `UnderReplicatedPartitions` é uma métrica importante e deve ser monitorada. Em um cluster MSK íntegro, essa métrica tem o valor igual a 0. Se for maior que zero, isso pode ocorrer por um dos motivos a seguir.

- Se `UnderReplicatedPartitions` estiver apresentando picos, o problema pode ser que o cluster não foi provisionado no tamanho correto para tratar o tráfego de entrada e saída. Consulte the section called “Práticas recomendadas para agentes padrão”.
- Se `UnderReplicatedPartitions` for consistentemente maior que 0, inclusive durante períodos de baixo tráfego, o problema pode ser que você tenha definido restrições ACLs que não concedem acesso ao tópico aos corretores. Para replicar partições, os agentes devem estar autorizados a READ (ler) e DESCRIBE (descrever) os tópicos. DESCRIBE é concedido por padrão com a autorização READ. Para obter informações sobre configuração ACLs, consulte Autorização e ACLs na documentação do Apache Kafka.

O cluster tem tópicos chamados __amazon_msk_canary e __amazon_msk_canary_state

Você pode ver que seu cluster do MSK tem um tópico com o nome `__amazon_msk_canary` e outro com o nome `__amazon_msk_canary_state`. Trata-se de tópicos internos que o Amazon MSK cria e usa para métricas de integridade e diagnóstico do cluster. Esses tópicos têm um tamanho insignificante e não podem ser excluídos.

Falha na replicação de partições

Certifique-se de não ter definido ACLs CLUSTER_ACTIONS.

Não é possível acessar o cluster que está com o acesso público ativado

Siga as etapas abaixo se o seu cluster estiver com o acesso público ativado, mas você ainda não conseguir acessá-lo pela Internet:

1. Certifique-se de que as regras de entrada do grupo de segurança do cluster tenham permissão para seu endereço IP e a porta do cluster. Para obter uma lista dos números de portas do cluster, consulte the section called “Informações de porta”. Certifique-se também de que as regras de saída do grupo de segurança permitam comunicações de saída. Para ter mais informações sobre grupos de segurança e suas regras de entrada e saída, consulte Grupos de segurança para sua VPC no Guia do usuário da Amazon VPC.
2. Certifique-se de que seu endereço IP e a porta do cluster tenham permissão nas regras de entrada da ACL da rede VPC do cluster. Ao contrário dos grupos de segurança, ACLs as redes não têm estado. Isso significa que você deve configurar as regras de entrada e saída. Nas regras de saída, permita que todo o tráfego (intervalo de portas: 0-65535) chegue ao seu endereço IP. Para obter mais informações, consulte Adicionar e excluir regras no Guia do usuário da Amazon VPC.
3. Verifique se você está usando a string bootstrap-brokers de acesso público para acessar o cluster. Um cluster do MSK com acesso público ativado tem duas strings distintas de agentes de inicialização, uma para acesso público e outra para acesso interno diretamente da AWS. Para obter mais informações, consulte the section called “Obtenha os corretores de bootstrap usando o AWS Management Console”.

Não é possível acessar o cluster de dentro AWS: problemas de rede

Se você tiver uma aplicação do Apache Kafka que não consiga se comunicar com êxito com um cluster do MSK, comece executando o teste de conectividade a seguir.

1. Use qualquer um dos métodos descritos em the section called “Obtenha os corretores de bootstrap” para obter os endereços dos agentes de bootstrap.
2. No comando a seguir, *`bootstrap-broker`* substitua por um dos endereços do broker que você obteve na etapa anterior. *`port-number`*Substitua por 9094 se o cluster estiver configurado para usar a autenticação TLS. Se o cluster não usar a autenticação TLS, *`port-number`* substitua por 9092. Execute o comando usando a máquina cliente.

```
telnet bootstrap-broker port-number
```

Em que o número da porta será:

- 9094 se o cluster estiver configurado para usar a autenticação TLS.
- 9092 se o cluster não usar a autenticação TLS.
- Um número de porta diferente será necessário se o acesso público estiver habilitado.

Execute o comando usando a máquina cliente.

3. Repita o comando anterior para todos os agentes de bootstrap.

Se a máquina cliente é capaz de acessar os agentes, isso significa que não há problemas de conectividade. Nesse caso, execute o comando a seguir para verificar se o cliente do Apache Kafka está configurado corretamente. Para obter*bootstrap-brokers*, use qualquer um dos métodos descritos emthe section called “Obtenha os corretores de bootstrap”. *topic*Substitua pelo nome do seu tópico.

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

Se o comando anterior for bem-sucedido, isso indica que o cliente está configurado corretamente. Se você ainda não consegue produzir e consumir de um aplicativo, depure o problema no nível do aplicativo.

Se a máquina cliente não consegue acessar os agentes, consulte as subseções a seguir para obter orientações baseadas na configuração da máquina cliente.

EC2 Cliente Amazon e cluster MSK na mesma VPC

Se a máquina cliente estiver na mesma VPC que o cluster do MSK, verifique se o grupo de segurança do cluster tem uma regra de entrada que aceite tráfego do grupo de segurança da máquina cliente. Para obter informações sobre como configurar essas regras, consulte Regras do grupo de segurança. Para ver um exemplo de como acessar um cluster de uma EC2 instância da Amazon que está na mesma VPC do cluster, consulte. the section called “Conceitos básicos”

EC2 Cliente Amazon e cluster MSK em diferentes VPCs

Se a máquina cliente e o cluster estiverem em duas partes diferentes VPCs, verifique o seguinte:

- Os dois VPCs são examinados.
- O status da conexão de emparelhamento está ativo.
- As tabelas de rotas dos dois VPCs estão configuradas corretamente.

Para obter informações sobre o emparelhamento de VPC, consulte Trabalhar com conexões de emparelhamento de VPC.

Cliente on-premises

No caso de um cliente local configurado para se conectar ao cluster MSK usando AWS VPN, verifique o seguinte:

- O status da conexão VPN é UP. Para obter informações sobre como verificar o status da conexão VPN, consulte Como verificar o status atual do meu túnel VPN?.
- A tabela de rotas da VPC do cluster contém a rota para um CIDR on-premises, cujo destino tem o formato `Virtual private gateway(vgw-xxxxxxxx)`.
- O grupo de segurança do cluster do MSK permite tráfego na porta 2181, na porta 9092 (se o cluster aceitar tráfego em texto simples) e na porta 9094 (se o cluster aceitar tráfego com criptografia TLS).

Para obter mais orientações sobre AWS VPN solução de problemas, consulte Solução de problemas do Client VPN.

AWS Direct Connect

Se o cliente usar AWS Direct Connect, consulte Solução de problemas AWS Direct Connect.

Se as orientações para a solução de problemas anteriores não resolverem a situação, certifique-se de que nenhum firewall esteja bloqueando o tráfego de rede. Para depuração adicional, use ferramentas como `tcpdump` e `Wireshark` para analisar o tráfego e garantir que ele esteja alcançando o cluster do MSK.

Falha na autenticação: muitas conexões

O erro `Failed authentication ... Too many connects` indica que um agente está se protegendo porque um ou mais clientes do IAM estão tentando se conectar a ele em um ritmo agressivo. Para ajudar os agentes a aceitarem uma taxa maior de novas conexões do IAM, você pode aumentar o parâmetro de configuração `reconnect.backoff.ms`.

Para saber mais sobre os limites de taxa para novas conexões por agente, consulte a página Cota do Amazon MSK.

Falha na autenticação: sessão muito curta

O `Failed authentication ... Session too short` erro ocorre quando seu cliente tenta se conectar a um cluster usando credenciais do IAM que estão prestes a expirar. Certifique-se de verificar como suas credenciais do IAM estão sendo atualizadas. Provavelmente, as credenciais estão sendo substituídas muito perto da expiração da sessão, o que causa problemas no servidor e falhas de autenticação.

MSK com tecnologia sem servidor: falha na criação do cluster

Se você tentar criar um cluster do MSK com a tecnologia sem servidor e o fluxo de trabalho falhar, talvez você não tenha permissão para criar um endpoint da VPC. Verifique se o administrador concedeu permissão para você criar um endpoint da VPC permitindo a ação `ec2:CreateVpcEndpoint`.

Para obter uma lista completa das permissões necessárias para realizar todas as ações do Amazon MSK, consulte AWS política gerenciada: Amazon MSKFull Access.

Não é possível atualizar KafkaVersionsList na configuração do MSK

Quando você atualiza a KafkaVersionsListpropriedade no AWS::MSK::Configurationrecurso, a atualização falha com o seguinte erro.

```
Resource of type 'AWS::MSK::Configuration' with identifier '<identifierName>' already
 exists.
```

Ao atualizar a `KafkaVersionsList` propriedade, AWS CloudFormation recria uma nova configuração com a propriedade atualizada antes de excluir a configuração antiga. A atualização da AWS CloudFormation pilha falha porque a nova configuração usa o mesmo nome da

configuração existente. Essa atualização requer uma substituição de recursos. Para atualizar com êxitoKafkaVersionsList, você também deve atualizar a propriedade Name na mesma operação.

Além disso, se sua configuração estiver vinculada a qualquer cluster criado usando o AWS Management Console ou AWS CLI, adicione o seguinte ao seu recurso de configuração para evitar tentativas malsucedidas de exclusão de recursos.

```
UpdateReplacePolicy: Retain
```

Depois que a atualização for bem-sucedida, acesse o console do Amazon MSK e exclua a configuração antiga. Para obter informações sobre configurações do MSK, consulte Configuração provisionada do Amazon MSK.

Melhores práticas para corretores Standard e Express

Esta seção descreve as melhores práticas a serem seguidas para corretores Standard e corretores Express. Para obter informações sobre as práticas recomendadas do Replicador do Amazon MSK, consulte Práticas recomendadas para usar o replicador do MSK.

Tópicos

- Práticas recomendadas para agentes padrão
- Melhores práticas para corretores Express
- Práticas recomendadas para clientes Apache Kafka

Práticas recomendadas para agentes padrão

Este tópico descreve algumas práticas recomendadas para seguir ao usar o Amazon MSK. Para obter informações sobre as práticas recomendadas do Replicador do Amazon MSK, consulte Práticas recomendadas para usar o replicador do MSK.

Considerações do cliente

A disponibilidade e o desempenho do seu aplicativo dependem não apenas das configurações do lado do servidor, mas também das configurações do cliente.

- Configurar seus clientes para alta disponibilidade. Em um sistema distribuído como o Apache Kafka, garantir a alta disponibilidade é crucial para manter uma infraestrutura de mensagens

confiável e tolerante a falhas. Os agentes ficarão offline em eventos planejados e não planejados, por exemplo, atualizações, aplicação de patches, falhas de hardware e problemas de rede. Um cluster do Kafka é tolerante a um agente offline, portanto, os clientes Kafka também devem lidar perfeitamente com o failover do agente. Veja os detalhes completos emPráticas recomendadas para clientes Apache Kafka.

- Certifique-se de que as strings de conexão do cliente incluam pelo menos um agente de cada zona de disponibilidade. Ter vários agentes na string de conexão de um cliente possibilita o failover quando um agente específico estiver offline para uma atualização. Para obter informações sobre como obter uma string de conexão com vários agentes, consulte Obter os agentes de bootstrap para um cluster do Amazon MSK.
- Execute testes de desempenho para verificar se as configurações do seu cliente permitem que você atinja seus objetivos de desempenho.

Considerações do servidor

Dimensione seu cluster adequadamente: número de partições por agente padrão

A tabela a seguir mostra o número recomendado de partições (incluindo partições líderes e seguidoras) por agente padrão. O número recomendado de partições não é imposto e é uma prática recomendada para cenários em que você está enviando tráfego por todas as partições de tópicos provisionadas.

Tamanho do agente	Número recomendado de partições (incluindo partições líderes e seguidoras) por agente	Número máximo de partições que suportam operações de atualização
`kafka.t3.small`	300	300
`kafka.m5.large` ou `kafka.m5.xlarge`	1000	1500
`kafka.m5.2xlarge`	2000	3000
`kafka.m5.4xlarge` , `kafka.m5.8xlarge` , `kafka.m5.12xlarge` ,	4000	6000

Tamanho do agente	Número recomendado de partições (incluindo partições líderes e seguidoras) por agente	Número máximo de partições que suportam operações de atualização
`kafka.m5.16xlarge` ou `kafka.m5.24xlarge`		
`kafka.m7g.large` ou `kafka.m7g.xlarge`	1000	1500
`kafka.m7g.2xlarge`	2000	3000
`kafka.m7g.4xlarge` , `kafka.m7g.8xlarge` , `kafka.m7g.12xlarge` ou `kafka.m7g.16xlarge`	4000	6000

Se você tem casos de uso de partição alta e baixa taxa de transferência em que tem um número maior de partições, mas não está enviando tráfego para todas as partições, você pode empacotar mais partições por agente, desde que tenha realizado testes e testes de desempenho suficientes para validar se o cluster permanece íntegro com o maior número de partições. Se o número de partições por agente exceder o valor máximo permitido e seu cluster ficar sobrecarregado, você será impedido de realizar as seguintes operações:

- Atualizar a configuração do cluster
- Atualizar o cluster para um tamanho de agente menor
- Associar um AWS Secrets Manager segredo do a um cluster que tenha autenticação SASL/SCRAM

Um grande número de partições também pode resultar na falta de métricas do Kafka na extração de dados do Prometheus. CloudWatch

Para obter orientações sobre como escolher o número de partições, consulte Apache Kafka Supports 200K Partitions Per Cluster. Também recomendamos que você execute seu próprio teste para determinar o tipo certo para os agentes. Para obter mais informações sobre os diferentes tamanhos de agentes, consulte the section called “Tipos de agente”.

Dimensione seu cluster adequadamente: número de agentes padrão por cluster

Para determinar o número certo de agentes padrão para seu cluster do MSK Provisioned e entender os custos, consulte a planilha Preço e dimensionamento do MSK. Essa planilha fornece uma estimativa para dimensionar um cluster do MSK e os custos associados do Amazon MSK em relação a um cluster do Apache Kafka semelhante, autogerenciado e baseado no Apache Kafka. EC2 Para obter mais informações sobre os parâmetros de entrada na planilha, passe o mouse sobre as descrições dos parâmetros. As estimativas fornecidas por essa planilha são conservadoras e fornecem um ponto de partida para um novo cluster do MSK. O desempenho, o tamanho e os custos do cluster dependerão do seu caso de uso e recomendamos que você os verifique com testes reais.

Para entender como a infraestrutura subjacente afeta o desempenho do Apache Kafka, consulte Práticas recomendadas para dimensionar corretamente seus clusters do Apache Kafka a fim de otimizar o desempenho e custo no blog Big Data. AWS A postagem do blog fornece informações sobre como dimensionar seus clusters para atender aos requisitos de throughput, disponibilidade e latência. Ela também fornece respostas para perguntas, como quando você deve aumentar a escala verticalmente ou horizontalmente, além de orientações sobre como verificar continuamente o tamanho dos seus clusters de produção. Para obter informações sobre clusters baseados em armazenamento hierárquico, consulte Melhores práticas para executar cargas de trabalho de produção usando o armazenamento hierárquico do Amazon MSK.

Otimizar o throughput do cluster para instâncias m5.4xl, m7g.4xl ou maiores

Ao usar instâncias m5.4xl, m7g.4xl ou maiores, você pode otimizar o throughput do cluster do MSK Provisioned ajustando as configurações num.io.threads e num.network.threads.

Num.io.threads é o número de threads que um agente padrão usa para processar solicitações. Adicionar mais threads, até o número de núcleos de CPU compatível com o tamanho da instância, pode ajudar a melhorar o throughput do cluster.

Num.network.threads é o número de threads que o agente padrão usa para receber todas as solicitações recebidas e retornar respostas. Os threads de rede colocam as solicitações recebidas em uma fila de solicitações para processamento por io.threads. Definir num.network.threads para a metade do número de núcleos de CPU compatível com o tamanho da instância permite o uso total do novo tamanho de instância.

> **Important**
>
> Não aumente num.network.threads sem antes aumentar num.io.threads, pois isso pode causar congestionamento relacionado à saturação da fila.

Configurações recomendadas

Tamanho da instância	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

Usar o Kafka mais recente AdminClient para evitar problemas de incompatibilidade de ID de tópico

O ID de um tópico é perdido (Erro: não corresponde ao ID do tópico para partição) quando você usa uma AdminClient versão do Kafka Kafka Kafka com o sinalizador para aumentar ou reatribuir partições de tópicos `--zookeeper` para um cluster do MSK Provisionado usando a versão 2.8.0 ou superior do Kafka. Observe que o sinalizador `--zookeeper` ficou obsoleto no Kafka 2.5 e foi removido desde o Kafka 3.0. Consulte Atualização para a versão 2.5.0 de qualquer versão entre 0.8.x e 2.4.x.

Para evitar incompatibilidade de ID de tópico, use um cliente do Kafka versão 2.8.0 ou superior para operações administrativas do Kafka. Como alternativa, clientes 2.5 e superiores podem usar o sinalizador `--bootstrap-servers` em vez do sinalizador `--zookeeper`.

Criar clusters altamente disponíveis

Aplique as recomendações a seguir para que seus clusters do MSK permaneça altamente disponíveis durante uma atualização (p. ex., quando você estiver atualizando o tamanho do agente ou a versão do Apache Kafka) ou quando o Amazon MSK estiver substituindo um agente.

- Configure um cluster com três zonas de disponibilidade.
- Certifique-se de que o Replication factor (RF – Fator de replicação) seja pelo menos 3. Observe que um RF de 1 pode resultar em partições offline durante uma atualização contínua; e um RF de 2 pode resultar em perda de dados.
- Defina réplicas mínimas em sincronização (minISR) para, no máximo, RF - 1. Uma minISR igual ao RF pode impedir a produção no cluster durante uma atualização sem interrupção. Uma minISR de 2 permite que tópicos replicados de três vias estejam disponíveis quando uma réplica estiver offline.

Monitorar uso da CPU

O Amazon MSK recomenda veementemente que você mantenha a utilização da CPU de seus agentes (definida como`CPU User + CPU System`) abaixo de 60%. Isso garante que seu cluster retenha espaço suficiente na CPU para lidar com eventos operacionais, como falhas de intermediários, patches e atualizações contínuas.

O Apache Kafka pode redistribuir a carga da CPU entre os agentes no cluster quando necessário. Por exemplo, quando o Amazon MSK detecta e se recupera de uma falha do agente, ele realiza a manutenção automática, como a aplicação de patches. Da mesma forma, quando um usuário solicita uma alteração do tamanho do agente ou um upgrade de versão, o Amazon MSK inicia fluxos de trabalho contínuos que colocam um agente offline por vez. Quando os agentes com partições principais ficam offline, o Apache Kafka reatribui a liderança da partição para redistribuir o trabalho para outros agentes no cluster. Ao seguir essa prática recomendada, você garante espaço suficiente na CPU para tolerar esses eventos operacionais.

> **Note**
>
> Ao monitorar a utilização da CPU, esteja ciente de que o uso total da CPU inclui mais de `CPU User` e. `CPU System` Outras categorias, como`iowait`,`irq`, e `softirqsteal`, também contribuem para a atividade geral da CPU. Consequentemente, a CPU ociosa nem sempre é igual a. `100% - CPU User - CPU System`

Você pode usar a matemática CloudWatch métrica da Amazon para criar uma métrica composta (`CPU User + CPU System`) e definir um alarme para ser acionado quando o uso médio exceder 60%. Quando acionado, considere escalar o cluster usando uma das seguintes opções:

- Opção 1 (recomendada): atualize o tamanho do agente para o tamanho maior seguinte. Por exemplo, se o tamanho atual for `kafka.m5.large`, atualize o cluster para usar `kafka.m5.xlarge`. Lembre-se de que, ao atualizar o tamanho do agente no cluster, o Amazon MSK coloca os agentes offline de forma contínua e reatribui temporariamente a liderança da partição para outros agentes. Normalmente uma atualização de tamanho leva de 10 a 15 minutos por agente.
- Opção 2: se houver tópicos com todas as mensagens ingeridas de produtores que usam gravações de ida e volta (em outras palavras, as mensagens não recebem chaves e a ordenação não é importante para os consumidores), expanda seu cluster adicionando agentes. Também adicione partições aos tópicos existentes com o maior throughput. Em seguida, use `kafka-topics.sh --describe` para garantir que as partições recém-adicionadas sejam atribuídas aos novos agentes. O principal benefício dessa opção em comparação com a anterior é que você pode gerenciar recursos e custos de modo mais granular. Além disso, você pode usar essa opção se a carga da CPU exceder significativamente 60%, pois essa forma de escalabilidade normalmente não resulta em aumento de carga nos agentes existentes.
- Opção 3: expanda seu cluster do MSK Provisioned adicionando agentes e, em seguida, reatribua as partições existentes usando a ferramenta de reatribuição de partições chamada. `kafka-reassign-partitions.sh` No entanto, se você usar essa opção, o cluster precisará gastar recursos para replicar dados de um agente para outro após a reatribuição das partições. Em comparação com as duas opções anteriores, inicialmente isso pode aumentar significativamente a carga no cluster. Como resultado, o Amazon MSK não recomenda usar essa opção quando a utilização da CPU estiver acima de 70%, pois a replicação causará carga adicional da CPU e tráfego de rede. O Amazon MSK recomenda usar essa opção somente se as duas opções anteriores não forem viáveis.

Outras recomendações:

- Monitore a utilização total da CPU por agente como um indicador da distribuição de carga. Se os agentes tiverem uma utilização consistentemente desigual da CPU, isso pode ser um sinal de que a carga não está sendo distribuída uniformemente no cluster. Recomendamos o uso do Cruise Control para gerenciar continuamente a distribuição de carga por meio da atribuição de partições.
- Monitore a latência da produção e do consumo. A latência da produção e do consumo pode aumentar linearmente com a utilização da CPU.
- Intervalo de extração do JMX: se você habilitar o monitoramento aberto com o recurso Prometheus, recomenda-se usar um intervalo de extração de 60 segundos ou mais (scrape_interval: 60s) para a configuração do host do Prometheus (prometheus.yml). A redução do intervalo de coleta pode levar a um alto uso da CPU em seu cluster.

Monitorar o espaço em disco

Para evitar ficar sem espaço em disco para mensagens, crie um CloudWatch alarme que observe a `KafkaDataLogsDiskUsed` métrica. Quando o valor dessa métrica atingir ou exceder 85%, execute uma ou mais das seguintes ações:

- Use the section called “Escalabilidade automática para clusters”. Você também pode aumentar manualmente o armazenamento do agente, conforme descrito em the section called “Escalabilidade manual”.
- Reduza o período de retenção de mensagens ou o tamanho do log. Para obter informações sobre como fazer isso, consulte the section called “Ajustar os parâmetros de retenção de dados”.
- Exclua tópicos não utilizados.

Para obter informações sobre como configurar e usar alarmes, consulte Usando alarmes da Amazon CloudWatch . Para obter uma lista completa das métricas do Amazon MSK, consulte the section called “Monitorar um cluster”.

Ajustar os parâmetros de retenção de dados

Consumir mensagens não as remove do log. Para liberar espaço em disco regularmente, é possível especificar explicitamente um período de retenção, ou seja, por quanto tempo as mensagens permanecem no log. Também é possível especificar um tamanho do log de retenção. Quando o período de retenção ou o tamanho do log de retenção são atingidos, o Apache Kafka começa a remover segmentos inativos do log.

Para especificar uma política de retenção no nível do cluster, defina um ou mais dos seguintes parâmetros: `log.retention.hours`, `log.retention.minutes`, `log.retention.ms` ou `log.retention.bytes`. Para obter mais informações, consulte the section called “Configurações personalizadas do Amazon MSK”.

Também é possível especificar parâmetros de retenção no nível do tópico:

- Para especificar um período de retenção por tópico, use o comando a seguir.

```
kafka-configs.sh --bootstrap-server $bs --alter --entity-type topics --entity-
name TopicName --add-config retention.ms=DesiredRetentionTimePeriod
```

- Para especificar um tamanho de log de retenção por tópico, use o comando a seguir.

```
kafka-configs.sh --bootstrap-server $bs --alter --entity-type topics --entity-
name TopicName --add-config retention.bytes=DesiredRetentionLogSize
```

Os parâmetros de retenção especificados no nível do tópico têm precedência sobre os parâmetros no nível do cluster.

Como acelerar a recuperação de logs após um desligamento inadequado

Após um desligamento inadequado, um agente pode demorar um pouco para reiniciar, pois registra a recuperação em log. Por padrão, o Kafka usa apenas um thread por diretório de log para realizar essa recuperação. Por exemplo, se você tiver milhares de partições, a conclusão da recuperação do log pode levar horas. Para acelerar a recuperação do log, recomenda-se aumentar o número de threads usando a propriedade de configuração `num.recovery.threads.per.data.dir`. É possível defini-la com o número de núcleos de CPU.

Monitorar a memória do Apache Kafka

Recomendamos que você monitore a memória que o Apache Kafka usa. Caso contrário, o cluster pode ficar indisponível.

Para determinar quanta memória o Apache Kafka usa, você pode monitorar a métrica `HeapMemoryAfterGC`. `HeapMemoryAfterGC` é o percentual da memória total da pilha que está em uso após a coleta de resíduos. Recomendamos que você crie um CloudWatch alarme que seja acionado quando `HeapMemoryAfterGC` aumentar acima de 60%.

As etapas que você pode seguir para diminuir o uso da memória variam. Elas dependem da forma como você configura o Apache Kafka. Por exemplo, se você usar a entrega de mensagens transacionais, poderá diminuir o valor `transactional.id.expiration.ms` na configuração do Apache Kafka de `604800000` ms para `86400000` ms (de 7 dias para 1 dia). Isso diminui o espaço ocupado na memória de cada transação.

Não adicionar agentes que não são do MSK

Para ZooKeeper clusters provisionados do MSK, se você usar ZooKeeper comandos do Apache para adicionar agentes, esses agentes não serão adicionados ao cluster do MSK. O Apache conterá informações incorretas sobre o cluster. ZooKeeper Isso pode resultar em perda de dados. Para obter as operações de cluster provisionadas do MSK suportadas, consulte. the section called “Principais características e conceitos”

Ativar a criptografia em trânsito

Para obter informações sobre a criptografia em trânsito e como ativá-la, consulte the section called “Criptografia do Amazon MSK em trânsito”.

Reatribuir partições

Para mover partições para diferentes agentes no mesmo cluster provisionado pelo MSK, você pode usar a ferramenta de reatribuição de partições chamada. `kafka-reassign-partitions.sh` Recomendamos que você não reatribua mais de 10 partições em uma única `kafka-reassign-partitions` chamada para operações seguras. Por exemplo, após adicionar novos agentes para expandir um cluster, ou mover partições a fim de remover agentes, você pode rebalancear esse cluster reatribuindo partições aos novos agentes. Para obter informações sobre como adicionar agentes a um cluster do MSK Provisioned, consulte. the section called “Expandir um cluster” Para obter informações sobre como remover agentes a um cluster do MSK Provisioned, consulte. the section called “Remover um agente” Para obter informações sobre a ferramenta de reatribuição de partições, consulte Expanding your cluster na documentação do Apache Kafka.

Melhores práticas para corretores Express

Este tópico descreve algumas das melhores práticas a serem seguidas ao usar corretores Express. Os corretores expressos vêm pré-configurados para alta disponibilidade e durabilidade. Seus dados são distribuídos em três zonas de disponibilidade por padrão, a replicação é sempre definida como 3 e a réplica mínima em sincronização está sempre definida como 2. No entanto, ainda há alguns fatores a serem considerados para otimizar a confiabilidade e o desempenho do seu cluster.

Considerações do lado do cliente

A disponibilidade e o desempenho do seu aplicativo dependem não apenas das configurações do lado do servidor, mas também das configurações do cliente.

- Configure seus clientes para alta disponibilidade. Em um sistema distribuído como o Apache Kafka, garantir a alta disponibilidade é crucial para manter uma infraestrutura de mensagens confiável e tolerante a falhas. Os corretores ficarão off-line para eventos planejados e não planejados, por exemplo, atualizações, correções, falhas de hardware e problemas de rede. Um cluster do Kafka é tolerante a um agente offline, portanto, os clientes Kafka também devem lidar perfeitamente com o failover do agente. Veja os detalhes completos nas recomendações de melhores práticas para clientes do Apache Kafka.
- Execute testes de desempenho para verificar se as configurações do seu cliente permitem que você atinja seus objetivos de desempenho mesmo quando reiniciamos os corretores sob carga máxima. Você pode reinicializar os agentes em seu cluster a partir do console do MSK ou usando o MSK. APIs

Considerações do lado do servidor

Tópicos

Dimensione seu cluster adequadamente: número de agentes por cluster

É fácil escolher o número de corretores para seu cluster baseado no Express. Cada corretor Express vem com uma capacidade de transferência definida para entrada e saída. Você deve usar essa capacidade de taxa de transferência como principal meio de dimensionar seu cluster (e depois considerar outros fatores, como partição e contagem de conexões, discutidos abaixo).

Por exemplo, se seu aplicativo de streaming precisar MBps de 45% de capacidade de entrada (gravação) e 90 de saída de MBps dados (leitura), você pode simplesmente usar 3 corretores express.m7g.large para atender às suas necessidades de taxa de transferência. Cada corretora

express.m7g.large lidará com 15% MBps das entradas e 30% das saídas. MBps Consulte a tabela a seguir para ver nossos limites de taxa de transferência recomendados para cada tamanho de corretora Express. Se sua taxa de transferência exceder os limites recomendados, você poderá ter um desempenho degradado e deverá reduzir seu tráfego ou escalar seu cluster. Se sua taxa de transferência exceder os limites recomendados e atingir a cota por corretora, a MSK limitará o tráfego do seu cliente para evitar mais sobrecarga.

Você também pode usar nossa planilha de dimensionamento e preços do MSK para avaliar vários cenários e considerar outros fatores, como contagem de partições.

Taxa de transferência máxima recomendada por corretora

Tamanho da instância	Entrada () MBps	Saída () 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

Monitorar uso da CPU

Recomendamos que você mantenha a utilização total da CPU de seus corretores (definida como Usuário da CPU + Sistema da CPU) abaixo de 60%. Quando você tiver ao menos 40% da CPU total do seu cluster disponível, o Apache Kafka poderá redistribuir a carga da CPU entre os agentes no cluster quando necessário. Isso pode ser necessário devido a eventos planejados ou não planejados. Um exemplo de evento planejado é um upgrade da versão do cluster durante o qual o MSK atualiza os agentes em um cluster reiniciando-os um por vez. Um exemplo de evento não planejado é uma falha de hardware em uma corretora ou, na pior das hipóteses, uma falha de AZ em que todos os corretores em uma AZ são afetados. Quando os agentes com réplicas de líderes de partição ficam off-line, o Apache Kafka reatribui a liderança da partição para redistribuir o trabalho

para outros agentes no cluster. Seguindo essa prática recomendada, você pode garantir que tenha espaço suficiente de CPU em seu cluster para tolerar eventos operacionais como esses.

Você pode usar o uso de expressões matemáticas com CloudWatch métricas no Guia CloudWatch do usuário da Amazon para criar uma métrica composta que é Usuário da CPU + Sistema da CPU. Defina um alarme que seja acionado quando a métrica composta atingir uma utilização média de 60% da CPU. Quando esse alarme for acionado, escale o cluster usando uma das seguintes opções:

- Opção 1: atualize o tamanho do seu corretor para o próximo tamanho maior. Lembre-se de que, ao atualizar o tamanho do agente no cluster, o Amazon MSK coloca os agentes offline de forma contínua e reatribui temporariamente a liderança da partição para outros agentes.
- Opção 2: expanda seu cluster adicionando agentes e, em seguida, reatribuindo partições existentes usando a ferramenta de reatribuição de partições chamada. `kafka-reassign-partitions.sh`

Outras recomendações

- Monitore a utilização total da CPU por agente como um indicador da distribuição de carga. Se os corretores tiverem uma utilização consistentemente desigual da CPU, isso pode ser um sinal de que a carga não está distribuída uniformemente no cluster. Recomendamos o uso do Cruise Control para gerenciar continuamente a distribuição de carga por meio da atribuição de partições.
- Monitore a latência da produção e do consumo. A latência da produção e do consumo pode aumentar linearmente com a utilização da CPU.
- Intervalo de captura do JMX: se você ativar o monitoramento aberto com o recurso Prometheus, é recomendável usar um intervalo de captura de 60 segundos ou mais () para a configuração do host do Prometheus (). `scrape_interval: 60s` `prometheus.yml` A redução do intervalo de coleta pode levar a um alto uso da CPU em seu cluster.

Dimensione seu cluster corretamente: número de partições por agente Express

Se você tem casos de uso de partição alta e baixa taxa de transferência em que tem um número maior de partições, mas não está enviando tráfego para todas as partições, você pode empacotar mais partições por agente, desde que tenha realizado testes e testes de desempenho suficientes para validar se o cluster permanece íntegro com o maior número de partições. Se o número de partições por agente exceder o valor máximo permitido e seu cluster ficar sobrecarregado, você será impedido de realizar as seguintes operações:

- Atualizar a configuração do cluster
- Atualizar o cluster para um tamanho de agente menor
- Associe um AWS Secrets Manager segredo a um cluster que tenha SASL/SCRAM autenticação

Um cluster sobrecarregado com um grande número de partições também pode resultar na falta de métricas do Kafka na coleta de dados do CloudWatch Prometheus.

Para obter orientações sobre como escolher o número de partições, consulte Apache Kafka Supports 200K Partitions Per Cluster. Também recomendamos que você execute seu próprio teste para determinar o tipo certo para os agentes. Para obter mais informações sobre os diferentes tamanhos de agentes, consulte Tamanhos dos agentes do Amazon MSK.

Para obter informações sobre o número recomendado de partições (incluindo réplicas líder e seguidora) para cada Express broker, consulte. Cota de partição do Express Broker O número recomendado de partições não é imposto e é uma prática recomendada para cenários em que você está enviando tráfego em todas as partições de tópicos provisionadas.

Monitore a contagem de conexões

As conexões do cliente com seus corretores consomem recursos do sistema, como memória e CPU. Dependendo do mecanismo de autenticação, você deve monitorar para garantir que está dentro dos limites aplicáveis. Para processar novas tentativas em conexões com falha, você pode definir o parâmetro de configuração `reconnect.backoff.ms` no lado do cliente. Por exemplo, se você quiser que um cliente tente novamente as conexões após 1 segundo, `reconnect.backoff.ms` defina `1000` como. Para obter mais informações sobre como configurar novas tentativas, consulte a documentação do Apache Kafka.

Dimensão	Quota
Máximo de conexões TCP por agente (controle de acesso IAM)	3000
Máximo de conexões TCP por agente (IAM)	100 por segundo
Máximo de conexões TCP por agente (não IAM)	O MSK não impõe limites de conexão para autenticação que não seja do IAM. No entanto, você deve monitorar outras métricas, como uso de CPU e memória, para garantir que não

Dimensão	Quota
	sobrecarregue seu cluster devido ao excesso de conexões.

Reatribuir partições

Para mover partições para diferentes agentes no mesmo cluster provisionado pelo MSK, você pode usar a ferramenta de reatribuição de partições chamada. `kafka-reassign-partitions.sh` Recomendamos que você não reatribua mais de 20 partições em uma única `kafka-reassign-partitions` chamada para operações seguras. Por exemplo, após adicionar novos agentes para expandir um cluster, ou mover partições a fim de remover agentes, você pode rebalancear esse cluster reatribuindo partições aos novos agentes. Para obter informações sobre como adicionar agentes a um cluster provisionado pelo MSK, consulte. the section called “Expandir um cluster” Para obter informações sobre como remover agentes de um cluster provisionado pelo MSK, consulte. the section called “Remover um agente” Para obter informações sobre a ferramenta de reatribuição de partições, consulte Expanding your cluster na documentação do Apache Kafka.

Práticas recomendadas para clientes Apache Kafka

Ao trabalhar com o Apache Kafka e o Amazon MSK, é importante configurar corretamente o cliente e o servidor para obter performance e confiabilidade ideais. Este guia fornece recomendações sobre as práticas recomendadas de configuração do lado do cliente para o Amazon MSK.

Para obter informações sobre as práticas recomendadas do Replicador do Amazon MSK, consulte Práticas recomendadas para usar o replicador do MSK. Para obter as melhores práticas dos corretores Standard e Express, consulteMelhores práticas para corretores Standard e Express.

Tópicos

- Disponibilidade do cliente Apache Kafka
- Performance do cliente Apache Kafka
- Monitoramento de clientes Kafka

Disponibilidade do cliente Apache Kafka

Em um sistema distribuído como o Apache Kafka, garantir a alta disponibilidade é crucial para manter uma infraestrutura de mensagens confiável e tolerante a falhas. Os agentes ficarão offline em

eventos planejados e não planejados, como atualizações, aplicação de patches, falhas de hardware e problemas de rede. Um cluster do Kafka é tolerante a um agente offline, portanto, os clientes Kafka também devem lidar perfeitamente com o failover do agente. Para garantir a alta disponibilidade dos clientes Kafka, recomendamos essas práticas recomendadas.

Disponibilidade do produtor

- Defina `retries` para instruir o produtor a fazer uma nova tentativa de enviar mensagens com falha durante o failover do agente. Recomendamos um valor de número inteiro máximo ou um valor alto semelhante para a maioria dos casos de uso. Não fazer essa definição prejudicará a alta disponibilidade do Kafka.
- Defina `delivery.timeout.ms` para especificar o limite máximo para o tempo total entre o envio de uma mensagem e o recebimento de uma confirmação do agente. Isso deve refletir os requisitos da empresa de quanto tempo uma mensagem deve ser válida. Defina o limite de tempo bem alto para permitir novas tentativas suficientes para concluir a operação de failover. Recomendamos um valor de 60 segundos ou mais para a maioria dos casos de uso.
- Defina `request.timeout.ms` como o máximo que uma única solicitação deve esperar antes de uma tentativa de reenvio. Recomendamos um valor de dez segundos ou mais para a maioria dos casos de uso.
- Defina `retry.backoff.ms` para configurar o atraso entre as novas tentativas para evitar uma tempestade de novas tentativas e impacto na disponibilidade. Recomendamos um valor mínimo de 200 ms para a maioria dos casos de uso.
- Defina `acks=all` para configurar alta durabilidade. Isso deve estar em linha com uma configuração de servidor de `RF=3` e `min.isr=2` para garantir que todas as partições no ISR reconheçam a gravação. Durante um único agente offline, isso é `min.isr`, ou seja 2.

Disponibilidade do consumidor

- Defina `auto.offset.reset` como `latest` inicialmente para grupos de consumidores novos ou recriados. Isso evita o risco de adicionar carga de cluster ao consumir todo o tópico.
- Defina `auto.commit.interval.ms` ao usar `enable.auto.commit`. Recomendamos um valor mínimo de cinco segundos para a maioria dos casos de uso para evitar o risco de carga adicional.
- Implemente o tratamento de exceções no código de processamento de mensagens do consumidor para lidar com erros transitórios, por exemplo, disjuntor ou suspensão com recuo exponencial. Não fazê-lo pode resultar em falhas de aplicações, o que pode causar rebalanceamento excessivo.
- Defina `isolation.level` para controlar como ler mensagens transacionais:

Recomendamos sempre configurar `read_uncommitted` implicitamente por padrão. Isso está ausente em algumas implementações de clientes.

Recomendamos um valor de `read_uncommitted` ao usar o armazenamento em camadas.

- Configure `client.rack` para usar a leitura de réplica mais próxima. Recomendamos configurar `az id` a fim de minimizar os custos e a latência do tráfego de rede. Consulte Reduce network traffic costs of your Amazon MSK consumers with rack awareness.

Rebalanceamentos de consumidores

- Defina `session.timeout.ms` para um valor maior do que o tempo de inicialização de uma aplicação, incluindo qualquer instabilidade de inicialização implementada. Recomendamos um valor de 60 segundos para a maioria dos casos de uso.
- Defina `heartbeat.interval.ms` para ajustar a forma como o coordenador do grupo vê um consumidor como íntegro. Recomendamos um valor de 10 segundos para a maioria dos casos de uso.
- Defina um mecanismo de desligamento na aplicação para fechar completamente o consumidor no SIGTERM, em vez de confiar nos tempos limite da sessão para identificar quando um consumidor sai de um grupo. As aplicações Kstream podem ser definidas como `internal.leave.group.on.close` para um valor de `true`.
- Defina `group.instance.id` como um valor distinto dentro do grupo de consumidores. O ideal é um nome de host, task-id ou pod-id. Recomendamos sempre definir isso para comportamentos mais determinísticos e uma melhor correlação de logs de cliente e servidor durante a solução de problemas.
- Defina `group.initial.rebalance.delay.ms` para um valor de acordo com o tempo médio de implantação. Isso interrompe os rebalanceamentos contínuos durante a implantação.
- Defina `partition.assignment.strategy` para usar atribuidores fixos. Recomendamos `StickyAssignor` ou `CooperativeStickyAssignor`.

Performance do cliente Apache Kafka

Para garantir a alta performance de clientes Kafka, recomendamos essas práticas recomendadas.

Performance do produtor

- Defina `linger.ms` para controlar o tempo que um produtor espera até que um lote seja preenchido. Lotes menores são componentes computacionais caros para o Kafka, pois representam mais threads e operações de E/S ao mesmo tempo. Recomendamos os valores a seguir.

  Um valor mínimo de 5 ms para todos os casos de uso, incluindo baixa latência.

  Recomendamos um valor maior de 25 ms para a maioria dos casos de uso.

  Recomendamos nunca usar um valor de zero em casos de uso de baixa latência. (Um valor zero normalmente causa latência, independentemente da sobrecarga de E/S).
- Defina `batch.size` para controlar o tamanho do lote enviado ao cluster. Recomendamos aumentar para um valor de 64 KB ou 128 KB.
- Defina `buffer.memory` ao usar tamanhos de lotes maiores. Recomendamos um valor de 64 MB para a maioria dos casos de uso.
- Defina `send.buffer.bytes` para controlar o buffer TCP usado para receber bytes. Recomendamos um valor de -1 para permitir que o sistema operacional gerencie esse buffer ao executar um produtor em uma rede de alta latência.
- Defina compression.type para controlar a compactação dos lotes. Recomendamos que o lz4 ou o zstd seja executado em um produtor em uma rede de alta latência.

Performance do consumidor

- Defina `fetch.min.bytes` para controlar o tamanho mínimo de busca válido para reduzir o número de buscas e a carga do cluster.

  Recomendamos um valor mínimo de 32 bytes para todos os casos de uso.

  Recomendamos um valor maior de 128 bytes para a maioria dos casos de uso.
- Defina fetch.max.wait.ms para determinar quanto tempo o consumidor esperará antes que fetch.min.bytes seja ignorado. Recomendamos um valor de 1.000 ms para a maioria dos casos de uso.
- Recomendamos que o número de consumidores seja pelo menos igual ao número de partições para melhor paralelismo e resiliência. Em algumas situações, é possível ter menos consumidores do que o número de partições para tópicos de baixa taxa de transferência.

- Defina `receive.buffer.bytes` para controlar o buffer TCP usado para receber bytes. Recomendamos um valor de -1 para permitir que o sistema operacional gerencie esse buffer ao executar um consumidor em uma rede de alta latência.

Conexões de cliente

O ciclo de vida das conexões tem um custo computacional e de memória em um cluster do Kafka. Muitas conexões criadas ao mesmo tempo causam uma carga que pode afetar a disponibilidade de um cluster do Kafka. Esse impacto na disponibilidade geralmente pode fazer com que as aplicações criem ainda mais conexões, causando uma falha em cascata, resultando em uma interrupção total. Um grande número de conexões pode ser obtido quando criado a uma taxa razoável.

Recomendamos as seguintes mitigações para gerenciar as altas taxas de criação de conexão:

- Certifique-se de que o mecanismo de implantação de aplicações não reinicie todos os produtores e consumidores de uma só vez, mas de preferência em lotes menores.
- Na camada da aplicação, o desenvolvedor deve garantir que uma instabilidade aleatória (suspensão aleatória) seja executada antes de criar um cliente administrador, cliente produtor ou cliente consumidor.
- No SIGTERM, ao fechar a conexão, uma suspensão aleatória deve ser executada para garantir que nem todos os clientes Kafka sejam fechados ao mesmo tempo. A suspensão aleatória deve ocorrer dentro do tempo limite antes que o SIGKILL ocorra.

  Example Exemplo A (Java)

  ```
  sleepInSeconds(randomNumberBetweenOneAndX);
                          this.kafkaProducer = new KafkaProducer<>(this.props);
  ```

  Example Exemplo B (Java)

  ```
  Runtime.getRuntime().addShutdownHook(new Thread(() -> {
      sleepInSeconds(randomNumberBetweenOneAndTwentyFive);
      kafkaProducer.close(Duration.ofSeconds(5));
  });
  ```

- Na camada da aplicação, o desenvolvedor deve garantir que os clientes sejam criados somente uma vez por aplicação em um padrão singleton. Por exemplo, ao usar o Lambda, o cliente deve ser criado no escopo global e não no método de manipulador.

- Recomendamos que o número de conexões seja monitorado com o objetivo de ser estável. creation/close/shiftA conexão é normal durante as implantações e o failover do agente.

Monitoramento de clientes Kafka

Monitorar os clientes Kafka é crucial para manter a integridade e a eficiência do ecossistema do Kafka. Seja você administrador, desenvolvedor ou membro da equipe de operações do Kafka, habilitar métricas do lado do cliente é fundamental para entender o impacto nos negócios durante eventos planejados e não planejados.

Recomendamos monitorar as métricas do lado do cliente a seguir usando o mecanismo de captura de métricas de sua preferência.

Ao criar tíquetes de suporte com a AWS, inclua quaisquer valores anormais observados durante o incidente. Inclua também um exemplo de logs da aplicação cliente detalhando os erros (não avisos).

Métricas do produtor

- 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**
>
> Erros transitórios com novas tentativas não são motivo de preocupação, pois isso faz parte do protocolo do Kafka para lidar com problemas transitórios, como failover de líder ou retransmissões de rede. `record-send-rate`confirmará se os produtores ainda estão realizando novas tentativas.

Métricas do consumidor

- 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**
>
> Práticas altas taxas de busca e de confirmação causarão uma carga desnecessária no cluster. É ideal realizar solicitações em lotes maiores.

Métricas comuns

- connection-close-rate
- connection-creation-rate
- connection-count

> **Note**
>
> criação e encerramento altos de conexão causarão uma carga desnecessária no cluster.

O que é o MSK Sem Servidor?

> **Note**
>
> O MSK Serverless está disponível nas regiões Leste dos EUA (Ohio), Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Oregon), Canadá (Central), Ásia-Pacífico (Mumbai), Ásia-Pacífico (Singapura), Ásia-Pacífico (Sydney), Ásia-Pacífico (Tóquio), Ásia-Pacífico (Seul), Europa (Frankfurt), Europa (Estocolmo) Europa (Irlanda), Europa (Paris) e Europa (Londres).

O MSK Serverless é um tipo de cluster para o Amazon MSK que possibilita que você execute o Apache Kafka sem precisar gerenciar e escalar a capacidade do cluster. Ele provisiona e dimensiona automaticamente a capacidade enquanto gerencia as partições em seu tópico, permitindo que você transmita dados sem pensar em dimensionar ou escalar clusters corretamente. O MSK Serverless oferece um modelo de preço baseado em throughput, para que você pague somente pelo que for usado. Considere usar um cluster com tecnologia sem servidor se suas aplicações precisarem de capacidade de streaming sob demanda que aumente e diminua automaticamente.

O MSK Serverless é totalmente compatível com o Apache Kafka, então você pode usar qualquer aplicação cliente compatível para produzir e consumir dados. Ele também se integra com os seguintes serviços:

- AWS PrivateLink para fornecer conectividade privada
- AWS Identity and Access Management (IAM) para autenticação e autorização usando linguagens Java e não Java. Para obter instruções sobre como configurar clientes para o IAM, consulte Configurar clientes para controle de acesso do IAM.
- AWS Glue Registro de esquemas para gerenciamento de esquemas
- Amazon Managed Service for Apache Flink para processamento de stream com base em Apache Flink
- AWS Lambda para processamento de eventos

> **Note**
>
> O MSK Serverless exige controle de acesso do IAM para todos os clusters. As listas de controle de acesso do Apache Kafka (ACLs) não são suportadas. Para obter mais informações, consulte the section called “Controle de acesso do IAM”.

Para obter informações sobre cotas de serviço aplicáveis ao MSK Serverless, consulte the section called “Cota para clusters com tecnologia sem servidor”.

Para ajudar você a começar a usar clusters com a tecnologia sem servidor e saber mais sobre as opções de configuração e monitoramento de clusters com a tecnologia sem servidor, consulte o seguinte.

Tópicos

- Usar clusters do MSK Sem Servidor
- Propriedades de configuração de clusters do MSK Sem Servidor
- Monitorar clusters do MSK Sem Servidor

Usar clusters do MSK Sem Servidor

Este tutorial mostra um exemplo de como você pode criar um cluster do MSK Serverless, criar uma máquina cliente capaz de acessá-lo e usar o cliente para criar tópicos no cluster e gravar dados nesses tópicos. Este exercício não representa todas as opções que você pode escolher ao criar um cluster com a tecnologia sem servidor. Em diferentes partes deste exercício, escolhemos as opções padrão para facilitar. Isso não significa que são as únicas opções que funcionam para configurar um cluster com a tecnologia sem servidor. Você também pode usar a API AWS CLI ou a Amazon MSK. Para obter mais informações, consulte a Referência 2.0 da API do Amazon MSK.

Tópicos

- Criar um cluster do Amazon MSK Sem Servidor
- Criar um perfil do IAM para tópicos do cluster do MSK Sem Servidor
- Criar uma máquina cliente para acessar o cluster do MSK Sem Servidor
- Criar um tópico do Apache Kafka
- Produzir e consumir dados no MSK Sem Servidor
- Excluir recursos que você criou para o MSK Sem Servidor

Criar um cluster do Amazon MSK Sem Servidor

Nesta etapa, você executará duas tarefas. Primeiro, você cria um cluster do MSK Serverless com as configurações padrão. Em seguida, você reúne informações sobre o cluster. Essas são as

informações que você precisará em etapas posteriores ao criar um cliente capaz de enviar dados para o cluster.

Para criar um cluster com a tecnologia sem servidor

1. Faça login no AWS Management Console e abra o console Amazon MSK em https://console.aws.amazon.com/msk/casa.
2. Selecione Criar cluster.
3. Em Método de criação, deixe a opção Criação rápida selecionada. A opção Criação rápida permite criar um cluster com a tecnologia sem servidor com as configurações padrão.
4. Em Nome do cluster, insira um nome descritivo, como **`msk-serverless-tutorial-cluster`**.
5. Em Propriedades gerais do cluster, escolha Tecnologia sem servidor como o Tipo de cluster. Use os valores padrão para nos itens restantes das Propriedades gerais do cluster.
6. Observe a tabela em Todas as configurações do cluster. Essa tabela lista os valores padrão para configurações importantes, como rede e disponibilidade, e indica se você pode alterar cada configuração depois de criar o cluster. Para alterar uma configuração antes de criar o cluster, você deve escolher a opção Criação personalizada em Método de criação.

   > **Note**
   >
   > Você pode conectar clientes de até cinco diferentes VPCs com clusters MSK Serverless. Para ajudar as aplicações clientes a migrarem para outra zona de disponibilidade no caso de uma interrupção, você deve especificar pelo menos duas sub-redes em cada VPC.

7. Selecione Criar cluster.

Para reunir informações sobre o cluster

1. Na seção Resumo do cluster, escolha Exibir informações do cliente. Esse botão permanece esmaecido até que o Amazon MSK conclua a criação do cluster. Pode ser necessário esperar alguns minutos até o botão ficar ativo para poder usá-lo.
2. Copie a string sob o rótulo Endpoint. Essa é a string do seu servidor bootstrap.
3. Escolha a guia Properties (Propriedades).

4. Na seção Configurações de rede, copie as sub-redes e o grupo IDs de segurança e salve-as, pois você precisará dessas informações posteriormente para criar uma máquina cliente.
5. Escolha qualquer uma das sub-redes. Isso abrirá o console da Amazon VPC. Localize o ID da Amazon VPC associada à sub-rede. Salve esse ID da Amazon VPC para uso posterior.

Próxima etapa

Criar um perfil do IAM para tópicos do cluster do MSK Sem Servidor

Criar um perfil do IAM para tópicos do cluster do MSK Sem Servidor

Nesta etapa, você executará duas tarefas. A primeira tarefa será a criação de uma política do IAM que conceda acesso para criar tópicos no cluster e enviar dados para esses tópicos. A segunda tarefa será a criação de um perfil do IAM e a associação dessa política a ele. Em uma etapa posterior, criaremos uma máquina cliente que vai assumir esse perfil e usá-lo para criar um tópico no cluster e enviar dados para esse tópico.

Para criar uma política do IAM que permita criar tópicos e gravar neles

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Políticas.
3. Escolha Create Policy.
4. Escolha a guia JSON e substitua o JSON na janela do editor com o JSON a seguir.

   No exemplo a seguir, substitua o seguinte:

   - *region*com o código de Região da AWS onde você criou seu cluster.
   - Exemplo de ID da conta*123456789012*, com seu Conta da AWS ID.
   - *msk-serverless-tutorial-cluster*/*c07c74ea-5146-4a03-add1-9baa787a5b14-s3*e *msk-serverless-tutorial-cluster* com seu ID de cluster sem servidor e nome do tópico.

   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 obter instruções sobre como criar políticas seguras, consultethe section called “Controle de acesso do IAM”.

5. Escolha Próximo: etiquetas.
6. Selecione Próximo: revisar.
7. Em nome da política, insira um nome descritivo, como **msk-serverless-tutorial-policy**.
8. Selecione Criar política.

Para criar um perfil do IAM e associar a política a ele

1. No painel de navegação, escolha Perfis.
2. Selecione Criar perfil.
3. Em Casos de uso comuns, escolha e EC2, em seguida, escolha Avançar: Permissões.

4. Na caixa de pesquisa, insira o nome da política que você criou anteriormente para este tutorial. Em seguida, marque a caixa à esquerda da política.
5. Escolha Próximo: etiquetas.
6. Selecione Próximo: revisar.
7. Em nome do perfil, insira um nome descritivo, como **`msk-serverless-tutorial-role`**.
8. Selecione Criar perfil.

Próxima etapa

Criar uma máquina cliente para acessar o cluster do MSK Sem Servidor

Criar uma máquina cliente para acessar o cluster do MSK Sem Servidor

Nesta etapa, você executará duas tarefas. A primeira tarefa é criar uma EC2 instância da Amazon para usar como uma máquina cliente Apache Kafka. A segunda tarefa é instalar as ferramentas Java e Apache Kafka na máquina.

Como criar uma máquina cliente

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. Escolha Iniciar instância.
3. Insira um Nome descritivo para sua máquina cliente, como **`msk-serverless-tutorial-client`**.
4. Deixe a opção AMI do Amazon Linux 2 (HVM) – Kernel 5.10, tipo de volume SSD selecionada para Tipo de imagem de máquina da Amazon (AMI).
5. Deixe o tipo de instância t2.micro selecionado.
6. Na seção Par de chaves, escolha Criar um novo par de chaves. Insira **`MSKServerlessKeyPair`** para Nome do par de chaves. Em seguida, escolha Baixar o par de chaves. Se preferir, use um par de chaves existente.
7. Em Configurações de rede, escolha Editar.
8. Em VPC, insira o ID da nuvem privada virtual (VPC) para o seu cluster com a tecnologia sem servidor. Trata-se da VPC baseada no serviço da Amazon VPC, cujo ID você salvou após a criação do cluster.
9. Em Sub-rede, escolha a sub-rede cujo ID você salvou depois de criar o cluster.

10. Em Firewall (grupos de segurança), selecione o grupo de segurança associado ao cluster. Esse valor funcionará se esse grupo de segurança tiver uma regra de entrada que permita tráfego do grupo de segurança para ele. Com essa regra, os membros do mesmo grupo de segurança podem se comunicar entre eles. Para obter mais informações, consulte Regras de grupos de segurança no Guia do desenvolvedor da Amazon VPC.
11. Expanda a seção Detalhes avançados e escolha o perfil do IAM que você criou na Criar um perfil do IAM para tópicos do cluster do MSK Sem Servidor.
12. Escolha Executar.
13. No painel de navegação à esquerda, selecione Instâncias. Em seguida, escolha a caixa de seleção na linha que representa sua EC2 instância Amazon recém-criada. Deste ponto em diante, chamamos essa instância de máquina cliente.
14. Escolha Conectar e siga as instruções para se conectar à máquina cliente.

Para configurar as ferramentas do cliente Apache Kafka na máquina cliente

1. Para instalar o Java, execute o seguinte comando na máquina cliente:

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

2. Para obter as ferramentas do Apache Kafka necessárias para criar tópicos e enviar dados, execute os seguintes 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**
>
> Depois de extrair o arquivo do Kafka, certifique-se de que os scripts no `bin` diretório tenham as permissões de execução adequadas. Para fazer isso, execute o comando a seguir.
>
> ```
> chmod +x kafka_2.12-2.8.1/bin/*.sh
> ```

3. Acesse o diretório `kafka_2.12-2.8.1/libs` e execute o seguinte comando para baixar o arquivo JAR do IAM do Amazon MSK. O JAR do IAM do Amazon MSK permite que a máquina cliente acesse o cluster.

   ```
   wget https://github.com/aws/aws-msk-iam-auth/releases/download/v2.3.0/aws-msk-iam-auth-2.3.0-all.jar
   ```

   Usando esse comando, você também pode baixar outras versões ou versões mais recentes do arquivo JAR IAM do Amazon MSK.

4. Acesse o diretório `kafka_2.12-2.8.1/bin`. Copie e cole as seguintes configurações de propriedade em um novo arquivo. Nomeie e salve o arquivo 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
   ```

Próxima etapa

Criar um tópico do Apache Kafka

Criar um tópico do Apache Kafka

Nesta etapa, você usa a máquina cliente criada anteriormente para criar um tópico no cluster com tecnologia sem servidor.

Tópicos

- Configurando seu ambiente para criar tópicos
- Criando um tópico e gravando dados nele

Configurando seu ambiente para criar tópicos

- Antes de criar um tópico, certifique-se de ter baixado o arquivo JAR do AWS MSK IAM para o diretório de instalação do Kafka. `libs/` Se você ainda não fez isso, execute o seguinte comando no diretório do `libs/` Kafka.

```
wget https://github.com/aws/aws-msk-iam-auth/releases/download/v2.3.0/aws-msk-iam-auth-2.3.0-all.jar
```

Esse arquivo JAR é necessário para a autenticação do IAM com seu cluster MSK Serverless.

- Ao executar comandos do Kafka, talvez seja necessário garantir que eles `classpath` incluam o arquivo JAR do AWS MSK IAM. Para isso, execute um dos seguintes procedimentos:
  - Defina a variável de `CLASSPATH` ambiente para incluir suas bibliotecas do Kafka, conforme mostrado no exemplo a seguir.

    ```
    export CLASSPATH=<path-to-your-kafka-installation>/libs/*:<path-to-your-kafka-installation>/libs/aws-msk-iam-auth-2.3.0-all.jar
    ```

  - Execute os comandos do Kafka usando o comando Java completo com explícito`classpath`, conforme mostrado no exemplo a seguir.

    ```
    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
    ```

Criando um tópico e gravando dados nele

1. No `export` comando a seguir, *`my-endpoint`* substitua pela string bootstrap-server que você salvou depois de criar o cluster. Em seguida, acesse o diretório `kafka_2.12-2.8.1/bin` na máquina cliente e execute o comando `export`.

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

2. Execute o comando a seguir para criar um tópico chamado `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
   ```

Próxima etapa

Produzir e consumir dados no MSK Sem Servidor

Produzir e consumir dados no MSK Sem Servidor

Nesta etapa, você produz e consome dados usando o tópico que criou na etapa anterior.

Como produzir e consumir mensagens

1. Execute o comando a seguir para criar um produtor de console.

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

2. Insira a mensagem que desejar e pressione Enter. Repita esta etapa duas ou três vezes. Sempre que você inserir uma linha e pressionar Enter, essa linha será enviada para o cluster como uma mensagem separada.
3. Mantenha a conexão com a máquina cliente aberta e abra uma segunda conexão separada com esse computador em uma nova janela.
4. Use sua segunda conexão com a máquina cliente para criar um consumidor no console executando o comando a seguir. *my-endpoint*Substitua pela string do servidor bootstrap que você salvou depois de criar o cluster.

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

   Você começará a ver as mensagens inseridas anteriormente quando usou o comando do produtor do console.
5. Insira mais mensagens na janela do produtor e observe-as aparecerem na janela do consumidor.

Se você encontrar `classpath` problemas ao executar esses comandos, certifique-se de executá-los no diretório correto. Além disso, certifique-se de que o JAR do AWS MSK IAM esteja no `libs` diretório. Como alternativa, você pode executar comandos do Kafka usando o comando Java completo com explícito`classpath`, conforme mostrado no exemplo a seguir.

```
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
```

Próxima etapa

Excluir recursos que você criou para o MSK Sem Servidor

Excluir recursos que você criou para o MSK Sem Servidor

Nesta etapa, você excluirá os recursos que criou neste tutorial.

Para excluir o cluster

1. Abra o console Amazon MSK em https://console.aws.amazon.com/msk/casa.
2. Na lista de clusters, escolha o cluster que criou para este tutorial.
3. Em Ações, escolha Excluir cluster.
4. Insira `delete` no campo e escolha Excluir.

Para interromper a máquina cliente

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. Na lista de EC2 instâncias da Amazon, escolha a máquina cliente que você criou para este tutorial.
3. Escolha Estado da instância e Encerrar instância.
4. Escolha Encerrar.

Para excluir a política e o perfil do IAM

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Perfis.
3. Na caixa de pesquisa, insira o nome do perfil do IAM que você criou para este tutorial.
4. Selecione o perfil de . Escolha Excluir perfil e confirme a exclusão.
5. No painel de navegação, escolha Políticas.
6. Na caixa de pesquisa, insira o nome da política que você criou para este tutorial.
7. Escolha a política para abrir a respectiva página de resumo. Na página Resumo da política, escolha Editar política.
8. Escolha Excluir.

Propriedades de configuração de clusters do MSK Sem Servidor

O Amazon MSK define propriedades de configuração do agente para clusters com tecnologia sem servidor. Você não pode alterar essas configurações de propriedades de configuração do agente. Porém, é possível definir ou modificar as propriedades de configuração no nível de tópico a seguir. Todas as outras propriedades de configuração no nível de tópico não são configuráveis.

Propriedade de configuração	Padrão	Editável	Valor máximo permitido
cleanup.policy	Excluir	Sim, mas somente no momento da criação do tópico	
compression.type	Produtor	Sim	
max.message.bytes	104858	Sim	8388608 (8 MiB)
message.timestamp.difference.max.ms	long.max	Sim	
message.timestamp.type	CreateTime	Sim	
retention.bytes	250 GiB	Sim	Ilimitado; defina-o como -1 para retenção ilimitada
retention.ms	7 dias	Sim	Ilimitado; defina-o como -1 para retenção ilimitada

Para definir ou modificar essas propriedades de configuração no nível de tópico, você pode usar as ferramentas de linhas de comandos do Apache Kafka. Consulte 3.2 Topic-level Configs na documentação oficial do Apache Kafka para obter mais informações e exemplos de como defini-las.

> **Note**
>
> Você não pode modificar a configuração de segment.bytes para tópicos no MSK Serverless. No entanto, um aplicativo do Kafka Streams pode tentar criar um tópico interno com um valor de configuração segment.bytes, que é diferente do que o MSK Serverless permitirá. Para obter informações sobre como configurar o Kafka Streams com o MSK Serverless, consulte. Usando o Kafka Streams com corretores MSK Express e MSK Serverless

Ao usar as ferramentas de linhas de comandos do Apache Kafka com o Amazon MSK Sem Servidor, certifique-se de concluir as etapas de 1 a 4 na seçãoTo set up Apache Kafka client tools on the client machine da documentação de conceitos básicos do Amazon MSK Sem Servidor. Além disso, você deve incluir o parâmetro `--command-config client.properties` nos comandos.

Por exemplo, o comando abaixo pode ser usado para modificar a propriedade de configuração do tópico retention.bytes para definir retenção 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
```

Neste exemplo, *`<bootstrap server string>`* substitua pelo endpoint do servidor bootstrap do seu cluster Amazon MSK Serverless e *`<topic_name>`* pelo nome do tópico que você deseja modificar.

O parâmetro `--command-config client.properties` garante que a ferramenta de linha de comandos do Kafka use as configurações apropriadas para se comunicar com o cluster do Amazon MSK Sem Servidor.

Monitorar clusters do MSK Sem Servidor

O Amazon MSK se integra à Amazon CloudWatch para que você possa coletar, visualizar e analisar métricas para seu cluster MSK Serverless. As métricas mostradas na tabela a seguir estão disponíveis para todos os clusters com tecnologia sem servidor. Como essas métricas são publicadas como pontos de dados individuais para cada partição no tópico, recomendamos visualizá-las como uma estatística “SUM” a fim de obter a visualização no nível de tópico.

O Amazon MSK publica `PerSec` métricas com CloudWatch uma frequência de uma vez por minuto. Isso significa que a estatística “SUM” para um período de um minuto representa com precisão os

dados por segundo para métricas `PerSec`. Para coletar dados por segundo por um período superior a um minuto, use a seguinte expressão CloudWatch matemática:`m1 * 60/PERIOD(m1)`.

Métricas disponíveis no nível de monitoramento DEFAULT

Name	Quando visível	Dimensões	Descrição
`BytesInPerSec`	Após um produtor gravar em um tópico	Nome do cluster, tópico	O número de bytes por segundo recebidos dos clientes. Essa métrica está disponível para cada tópico.
`BytesOutPerSec`	Após um grupo de consumidores consumir de um tópico	Nome do cluster, tópico	O número de bytes por segundo enviados aos clientes. Essa métrica está disponível para cada tópico.
`FetchMessageConversionsPerSec`	Após um grupo de consumidores consumir de um tópico	Nome do cluster, tópico	O número de conversões de mensagens de busca por segundo para o tópico.
`EstimatedMaxTimeLag`	Após um grupo de consumidores consumir de um tópico	Nome do cluster, grupo de consumidores, tópico	Uma estimativa de tempo da MaxOffsetLag métrica.
`MaxOffsetLag`	Após um grupo de consumidores consumir de um tópico	Nome do cluster, grupo de consumidores, tópico	O atraso máximo de deslocamento entre todas as partições em um tópico.
`MessagesInPerSec`	Após um produtor gravar em um tópico	Nome do cluster, tópico	O número de mensagens recebidas por segundo para o tópico.
`ProduceMessageConversionsPerSec`	Após um produtor gravar em um tópico	Nome do cluster, tópico	O número de conversões de mensagens de produção por segundo para o tópico.

Name	Quando visível	Dimensões	Descrição
`SumOffsetLag`	Após um grupo de consumidores consumir de um tópico	Nome do cluster, grupo de consumidores, tópico	O atraso de deslocamento agregado para todas as partições em um tópico.

Para visualizar as métricas do MSK Serverless

1. Faça login no AWS Management Console e abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, em Métricas, escolha Todas as métricas.
3. Nas métricas, pesquise o termo **`kafka`**.
4. Escolha AWS/Kafka/Nome do cluster, tópico ou AWS/Kafka/Nome do cluster, grupo de consumidores, tópico para ver métricas diferentes.

Saiba mais sobre o MSK Connect

O MSK Connect é um recurso do Amazon MSK que facilita o streaming de dados de e para os clusters do Apache Kafka. O MSK Connect usa as versões 2.7.1 ou 3.7.x do Kafka Connect, que são estruturas de código aberto para conectar clusters do Apache Kafka a sistemas externos, como bancos de dados, índices de pesquisa e sistemas de arquivos. Com o MSK Connect, você pode implantar conectores totalmente gerenciados criados para o Kafka Connect que movem dados para ou extraem dados de datastores populares, como Amazon S3 e Amazon Service. OpenSearch Você pode implantar conectores desenvolvidos por terceiros, como o Debezium, para transmitir logs de alterações de bancos de dados para um cluster do Apache Kafka ou implantar um conector existente sem alterações no código. Os conectores escalam automaticamente para se ajustar às mudanças na carga, e você paga apenas pelos recursos que usa.

Use conectores de origem para importar dados de sistemas externos para seus tópicos. Com conectores de coletor, você pode exportar dados de seus tópicos para sistemas externos.

O MSK Connect é compatível com conectores para qualquer cluster do Apache Kafka com conectividade com uma Amazon VPC, seja um cluster do MSK ou um cluster do Apache Kafka hospedado de maneira independente.

O MSK Connect monitora continuamente a integridade e o estado de entrega dos conectores, corrige e gerencia o hardware subjacente e dimensiona automaticamente a escala dos conectores para corresponder às mudanças no throughput.

Para começar a usar o MSK Connect, consulte the section called “Começar”.

Para saber mais sobre os AWS recursos que você pode criar com o MSK Connectthe section called “Saiba mais sobre conectores”, consultethe section called “Criar plug-ins personalizados”, e. the section called “Saiba mais sobre os operadores do MSK Connect”

Para obter informações sobre a API do MSK Connect, consulte a Referência de API do Amazon MSK Connect.

Benefícios de usar o Amazon MSK Connect

O Apache Kafka é uma das plataformas de streaming de código aberto mais amplamente adotadas para ingerir e processar fluxos de dados em tempo real. Com o Apache Kafka, você pode desacoplar e escalar de forma independente as aplicações que produzem e consomem dados.

O Kafka Connect é um componente importante da criação e execução de aplicações de streaming com o Apache Kafka. O Kafka Connect fornece uma maneira padronizada de mover dados entre o Kafka e sistemas externos. O Kafka Connect é altamente escalável e pode lidar com grandes volumes de dados. Ele fornece um conjunto avançado de operações e ferramentas de API para configurar, implantar e monitorar conectores que movem dados entre tópicos do Kafka e sistemas externos. Você pode usar essas ferramentas para personalizar e ampliar a funcionalidade do Kafka Connect para atender às necessidades específicas da aplicação de streaming.

Você pode enfrentar desafios ao operar clusters do Apache Kafka Connect por conta própria, ou ao tentar migrar aplicações de código aberto do Apache Kafka Connect para a AWS. Esses desafios incluem o tempo necessário para configurar a infraestrutura e implantar aplicações, obstáculos de engenharia ao configurar clusters autogerenciados do Apache Kafka Connect e sobrecarga operacional administrativa.

Para enfrentar esses desafios, recomendamos usar o Amazon Managed Streaming for Apache Kafka Connect (Amazon MSK Connect) para migrar as aplicações do Apache Kafka Connect de código aberto para a AWS. O Amazon MSK Connect simplifica o uso do Kafka Connect para transmitir dados de e para entre clusters do Apache Kafka e sistemas externos, como bancos de dados, índices de pesquisa e sistemas de arquivos.

Veja abaixo alguns benefícios de migrar para o Amazon MSK Connect:

- Eliminação da sobrecarga operacional: o Amazon MSK Connect elimina a carga operacional associada à aplicação de patches, provisionamento e escalabilidade dos clusters do Apache Kafka Connect. O Amazon MSK Connect monitora continuamente a integridade dos clusters do Connect e automatiza a aplicação de patches e as atualizações de versão sem causar interrupções nas workloads.
- Reinício automático das tarefas do Connect: o Amazon MSK Connect pode recuperar automaticamente tarefas com falha para reduzir as interrupções na produção. As falhas nas tarefas podem ser causadas por erros temporários, como a violação do limite de conexão TCP do Kafka e o rebalanceamento de tarefas quando novos operadores se juntam ao grupo de consumidores para conectores de coletor.
- Escalabilidade horizontal e vertical automática: o Amazon MSK Connect permite que a aplicação de conectores seja escalada automaticamente para ser compatível com maiores taxas de transferência. O Amazon MSK Connect gerencia a escalabilidade para você. Você só precisa especificar o número de operadores no grupo do Auto Scaling e os limites de utilização.o Você pode usar a operação da `UpdateConnector` API Amazon MSK Connect para aumentar ou reduzir verticalmente o v CPUs entre 1 e 8 v CPUs para suportar a taxa de transferência variável.

- Conectividade de rede privada — O Amazon MSK Connect se conecta de forma privada aos sistemas de origem e coletor usando nomes AWS PrivateLink DNS privados.

Conceitos básicos sobre o MSK Connect

Este é um step-by-step tutorial que usa o AWS Management Console para criar um cluster MSK e um conector de coletor que envia dados do cluster para um bucket S3.

Tópicos

Configurar os recursos necessários para o MSK Connect

Nesta etapa, você cria os seguintes recursos necessários para esse cenário inicial:

- Um bucket do Amazon S3 para servir como destino que recebe dados do conector.
- Um cluster do MSK para o qual você enviará dados. Em seguida, o conector lerá os dados desse cluster e os enviará para o bucket S3 de destino.
- Uma política do IAM que contém as permissões para gravar no bucket S3 de destino.
- Uma perfil do IAM que permite ao conector gravar no bucket do S3 de destino. Você adicionará a política do IAM que você criou a essa função.
- Um endpoint da Amazon VPC para possibilitar o envio de dados da Amazon VPC que tem o cluster e o conector para o Amazon S3.

Para criar um bucket do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em. https://console.aws.amazon.com/s3/
2. Escolha Criar bucket.

3. Para o nome do bucket, insira um nome descritivo, como `amzn-s3-demo-bucket-mkc-tutorial`.
4. Role para baixo e escolha Criar bucket.
5. Na lista de buckets, escolha o bucket recém-criado.
6. Selecione Criar pasta.
7. Digite `tutorial` para o nome da pasta, depois role para baixo e escolha Criar pasta.

Para criar um cluster

1. Abra o console Amazon MSK em https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. No painel esquerdo, em Clusters do MSK, escolha Clusters.
3. Selecione Criar cluster.
4. Em Método de criação, escolha Criação personalizada.
5. Insira **`mkc-tutorial-cluster`** para o nome do cluster.
6. Em Tipo de cluster, escolha Provisionado.
7. Escolha Próximo.
8. Em Rede, escolha uma Amazon VPC. Em seguida, selecione as zonas de disponibilidade e as sub-redes que deseja usar. Lembre-se IDs da Amazon VPC e das sub-redes que você selecionou porque precisa delas posteriormente neste tutorial.
9. Escolha Próximo.
10. Em Métodos de controle de acesso, verifique se somente o Acesso não autenticado está selecionado.
11. Em Criptografia, certifique-se de que somente Texto simples esteja selecionado.
12. Continue com o assistente e escolha Criar cluster. Você será redirecionado para a página detalhes do cluster. Nessa página, em Grupos de segurança aplicados, encontre o ID do grupo de segurança. Lembre-se desse ID porque você precisará dele posteriormente neste tutorial.

Para criar uma política do IAM com permissões para gravar no bucket do S3

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Políticas.
3. Selecione Criar política.

4. No Editor de políticas, escolha JSON e, em seguida, substitua o JSON na janela do editor pelo seguinte JSON.

   No exemplo a seguir, *<amzn-s3-demo-bucket-my-tutorial>* substitua pelo nome do seu bucket do 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 obter instruções sobre como criar políticas seguras, consultethe section called “Controle de acesso do IAM”.

5. Escolha Próximo.

6. Na página Revisar e criar, faça o seguinte:

a. Em Nome da política, insira um nome descritivo, como **`mkc-tutorial-policy`**.

b. Em Permissões definidas nesta política, revise e and/or edite as permissões definidas em sua política.

c. (Opcional) Para ajudar a identificar, organizar ou pesquisar a política, escolha Adicionar nova tag para adicionar tags como pares de valores-chave. Por exemplo, adicione uma tag à sua política com o par de **`Environment`** valores-chave e. **`Test`**

   Para obter mais informações sobre o uso de tags, consulte Tags para AWS Identity and Access Management recursos no Guia do usuário do IAM.

7. Selecione Criar política.

Para criar o perfil do IAM capaz de gravar no bucket de destino

1. No painel de navegação do console do IAM, escolha Roles e, em seguida, escolha Create role.
2. Na página Select trusted entity (Selecionar entidade confiável), faça o seguinte:

   a. Em Tipo de Entidade Confiável, escolha AWS service (Serviço da AWS).

   b. Para Serviço ou caso de uso, escolha S3.

   c. Em Caso de uso, escolha S3.

3. Escolha Próximo.
4. Na página Add permissions (Adicionar permissões), faça o seguinte:

   a. Na caixa de pesquisa em Políticas de permissões, insira o nome da política que você criou anteriormente para este tutorial. Por exemplo, .mkc-tutorial-policy Em seguida, escolha a caixa à esquerda do nome da política.

   b. (Opcional) Defina um limite de permissões. Esse é um atributo avançado que está disponível para perfis de serviço, mas não para perfis vinculados ao serviço. Para obter informações sobre como definir um limite de permissões, consulte Como criar funções e anexar políticas (console) no Guia do usuário do IAM.

5. Escolha Próximo.
6. Na página Name, review, and create (Nomear, revisar e criar), faça o seguinte:

   a. Em Nome da função, insira um nome descritivo, como **`mkc-tutorial-role`**.

> **Important**
>
> Quando nomear um perfil, observe o seguinte:
>
> - Os nomes das funções devem ser exclusivos dentro de você Conta da AWS e não podem ser diferenciados por maiúsculas e minúsculas.
>
> Por exemplo, não crie dois perfis denominados **`PRODROLE`** e **`prodrole`**. Quando usado em uma política ou como parte de um ARN, o nome de perfil diferencia maiúsculas de minúsculas. No entanto, quando exibido para os clientes no console, como durante o processo de login, o nome de perfil diferencia maiúsculas de minúsculas.
> - Não é possível editar o nome do perfil depois de criá-lo porque outras entidades podem referenciar o perfil.

b. (Opcional) Em Descrição, insira uma descrição para o perfil.

c. (Opcional) Para editar os casos de uso e as permissões da função, na Etapa 1: Selecionar entidades confiáveis ou Etapa 2: Adicionar seções de permissões, escolha Editar.

d. (Opcional) Para ajudar a identificar, organizar ou pesquisar a função, escolha Adicionar nova tag para adicionar tags como pares de valores-chave. Por exemplo, adicione uma tag à sua função com o par de **`ProductManager`** valores-chave e. **`John`**

 Para obter mais informações sobre o uso de tags, consulte Tags para AWS Identity and Access Management recursos no Guia do usuário do IAM.

7. Reveja a função e escolha Criar função.

Para permitir que o MSK Connect assuma o perfil

1. No console do IAM, em Gerenciamento de acesso no painel esquerdo, escolha Perfis.
2. Encontre e escolha o `mkc-tutorial-role`.
3. Na página Resumo do perfil, escolha a guia Relações de confiança.
4. Selecione Edit trust relationship (Editar relação de confiança).
5. Substitua a política de confiança existente pelo seguinte JSON.

JSON

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

6. Selecione Atualizar política de confiança.

Para criar um endpoint da Amazon VPC da VPC do cluster para o Amazon S3

1. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
2. No painel esquerdo, escolha Endpoints.
3. Escolha Criar endpoint.
4. Em Nome do serviço, escolha o serviço com.amazonaws.us-east-1.s3 e o tipo Gateway.
5. Escolha a VPC do cluster e, em seguida, selecione a caixa à esquerda da tabela de rotas associada às sub-redes do cluster.
6. Escolha Criar endpoint.

Próxima etapa

Criar plug-in personalizado

Criar plug-in personalizado

Um plug-in contém o código que define a lógica do conector. Nesta etapa, você criará um plug-in personalizado contendo o código para o Lenses Amazon S3 Sink Connector. Em uma etapa posterior, ao criar o conector do MSK, você especificará que seu código está nesse plug-in

personalizado. Você pode usar o mesmo plug-in para criar vários conectores do MSK com configurações diferentes.

Para criar o plug-in personalizado

1. Baixe o conector do S3.
2. Faça upload do arquivo ZIP para um bucket do S3 ao qual você tenha acesso. Para obter informações sobre como fazer upload de arquivos para o Amazon S3, consulte Carregar objetos no Guia do usuário do Amazon S3.
3. Abra o console do Amazon MSK em https://console.aws.amazon.com/msk/.
4. No painel esquerdo, expanda MSK Connect e escolha Plug-ins personalizados.
5. Escolha Criar plug-in personalizado.
6. Selecione Browse S3 (Navegar no S3).
7. Na lista de buckets, encontre o bucket no qual você fez o upload do arquivo ZIP e escolha-o.
8. Na lista de objetos no bucket, marque o botão de seleção à esquerda do arquivo ZIP e selecione o botão Escolher.
9. Insira `mkc-tutorial-plugin` para o nome do plug-in personalizado e escolha Criar plug-in personalizado.

Pode levar AWS alguns minutos para concluir a criação do plug-in personalizado. Quando o processo de criação estiver concluído, você verá a seguinte mensagem em um banner na parte superior da janela do navegador.

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

Próxima etapa

Criar a máquina cliente e o tópico do Apache Kafka

Criar a máquina cliente e o tópico do Apache Kafka

Nesta etapa, você cria uma EC2 instância da Amazon para usar como uma instância cliente Apache Kafka. Em seguida, você usará essa instância para criar um tópico no cluster.

Como criar uma máquina cliente

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. Selecione Iniciar instâncias.
3. Insira um Nome para sua máquina cliente, como **`mkc-tutorial-client`**.
4. Deixe a opção AMI do Amazon Linux 2 (HVM) – Kernel 5.10, tipo de volume SSD selecionada para Tipo de imagem de máquina da Amazon (AMI).
5. Escolha o tipo de instância t2.xlarge.
6. Na seção Par de chaves, escolha Criar um novo par de chaves. Digite **`mkc-tutorial-key-pair`** em Nome do par de chaves e, em seguida, escolha Baixar par de chaves. Se preferir, use um par de chaves existente.
7. Escolha Iniciar instância.
8. Escolha View Instances (Exibir instâncias). Na coluna Grupos de segurança, escolha o grupo de segurança que está associado à sua nova instância. Copie o ID do grupo de segurança e salve-o para usar posteriormente.

Para permitir que o cliente recém-criado envie dados para o cluster

1. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
2. No painel esquerdo, em SEGURANÇA, escolha Grupos de segurança). Na coluna ID do grupo de segurança, localize o grupo de segurança do cluster. Você salvou o ID desse grupo de segurança ao criar o cluster em the section called “Configurar os recursos necessários para o MSK Connect”. Escolha esse grupo de segurança marcando a caixa à esquerda de sua linha. Certifique-se de que nenhum outro grupo de segurança seja selecionado simultaneamente.
3. Na metade inferior da tela, escolha a guia Regras de entrada.
4. Escolha Editar regras de entrada.
5. Na parte inferior esquerda da tela, escolha Adicionar regra.
6. Na nova regra, escolha All traffic (Todo o tráfego) na coluna Type (Tipo). No campo à direita da coluna Origem, insira o ID do grupo de segurança da máquina cliente. Trata-se do ID do grupo de segurança que você salvou após criar a máquina cliente.
7. Selecione Salvar rules. Agora, seu cluster do MSK aceitará todo o tráfego do cliente criado no procedimento anterior.

Para criar um tópico

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. Na tabela de instâncias, escolha `mkc-tutorial-client`.
3. Na parte superior da tela, escolha Connect e siga as instruções para se conectar à instância.
4. Instale o Java na instância do cliente executando o seguinte comando:

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

5. Execute o comando a seguir para fazer download do Apache Kafka.

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

> **Note**
>
> Se quiser usar um local de espelhamento diferente do usado neste comando, você poderá escolher um local diferente no site do Apache.

6. Execute o comando a seguir no diretório onde você fez download do arquivo TAR na etapa anterior.

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

7. Acesse o diretório kafka_2.12-2.2.1.
8. Abra o console Amazon MSK em https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
9. No painel esquerdo, escolha Clusters e, em seguida, escolha o nome `mkc-tutorial-cluster`.
10. Escolha Exibir informações do cliente.
11. Copie a string de conexão em texto simples.
12. Selecione Concluído.
13. Execute o comando a seguir na instância do cliente (`mkc-tutorial-client`), *`bootstrapServerString`* substituindo-o pelo valor que você salvou ao visualizar as informações do cliente do cluster.

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

Se o comando tiver êxito, a seguinte mensagem será exibida: `Created topic mkc-tutorial-topic.`

Próxima etapa

Criar um conector

Criar um conector

Este procedimento descreve como criar um conector usando o AWS Management Console.

Para criar o conector

1. Faça login no AWS Management Console e abra o console Amazon MSK em https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. No painel esquerdo, expanda MSK Connect e escolha Conectores.
3. Escolha Criar conector.
4. Na lista de plug-ins, escolha `mkc-tutorial-plugin` e escolha Próximo.
5. Para o nome do conector, insira `mkc-tutorial-connector`.
6. Na lista de clusters, escolha `mkc-tutorial-cluster`.
7. Copie a seguinte configuração e cole no campo de configuração do conector.

   Certifique-se de substituir a região pelo código de Região da AWS onde você está criando o conector. Além disso, substitua o nome do bucket do *<amzn-s3-demo-bucket-my-tutorial>* Amazon S3 pelo nome do seu bucket no exemplo a seguir.

```
connector.class=io.confluent.connect.s3.S3SinkConnector
s3.region=us-east-1
format.class=io.confluent.connect.s3.format.json.JsonFormat
flush.size=1
schema.compatibility=NONE
tasks.max=2
topics=mkc-tutorial-topic
```

```
partitioner.class=io.confluent.connect.storage.partitioner.DefaultPartitioner
storage.class=io.confluent.connect.s3.storage.S3Storage
s3.bucket.name=<amzn-s3-demo-bucket-my-tutorial>
topics.dir=tutorial
```

8. Em Permissões de acesso, escolha `mkc-tutorial-role`.
9. Escolha Próximo. Na página Segurança, escolha Próximo novamente.
10. Na página Logs, escolha Próximo.
11. Em Revisar e criar, escolha Criar conector.

Próxima etapa

Enviar dados para o cluster do MSK

Enviar dados para o cluster do MSK

Nesta etapa, você envia dados para o tópico do Apache Kafka que você criou anteriormente e, em seguida, procura esses mesmos dados no bucket do S3 de destino.

Para enviar dados para o cluster do MSK

1. Na pasta `bin` da instalação do Apache Kafka na instância do cliente, crie um arquivo de texto chamado `client.properties` com o conteúdo a seguir.

```
security.protocol=SASL_SSL
sasl.mechanism=AWS_MSK_IAM
```

2. Execute o comando a seguir para criar um produtor de console. *`BootstrapBrokerString`*Substitua pelo valor obtido ao executar o comando anterior.

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

3. Insira a mensagem que desejar e pressione Enter. Repita esta etapa duas ou três vezes. Toda vez que você inserir uma linha e pressionar Enter, essa linha será enviada para o cluster do Apache Kafka como uma mensagem separada.
4. Verifique o bucket do Amazon S3 de destino para encontrar as mensagens que você enviou na etapa anterior.

Saiba mais sobre conectores

Um conector integra sistemas externos e serviços da Amazon ao Apache Kafka, copiando continuamente dados de streaming de uma fonte de dados para o cluster do Apache Kafka ou copiando continuamente os dados do cluster para um coletor de dados. Antes de entregar os dados a um destino, um conector também pode executar uma lógica leve, como transformação, conversão de formato ou filtragem de dados. Os conectores de origem extraem dados de uma fonte de dados e os enviam para o cluster, enquanto os conectores coletam dados do cluster e os enviam para um coletor de dados.

O diagrama a seguir mostra a arquitetura de um conector. Um operador é um processo de máquina virtual Java (JVM) que executa a lógica do conector. Cada operador cria um conjunto de tarefas que são executadas em threads paralelos e fazem o trabalho de copiar os dados. As tarefas não armazenam o estado e, portanto, podem ser iniciadas, interrompidas ou reiniciadas a qualquer momento para fornecer um pipeline de dados resiliente e escalável.

Saiba mais sobre a capacidade de conectores

A capacidade total de um conector depende do número de trabalhadores que o conector tem, bem como do número de MSK Connect Units (MCUs) por trabalhador. Cada MCU representa 1 vCPU de computação e 4 GiB de memória. A memória da MCU pertence à memória total de uma instância de trabalho e não à memória de pilha em uso.

Os operadores do MSK Connect consomem endereços IP nas sub-redes fornecidas pelo cliente. Cada operador usa um endereço IP de uma das sub-redes fornecidas pelo cliente. Você deve garantir que tenha endereços IP disponíveis suficientes nas sub-redes fornecidas a uma CreateConnector solicitação para considerar a capacidade especificada, especialmente ao escalar automaticamente conectores em que o número de trabalhadores pode flutuar.

Para criar um conector, você deve escolher entre um dos dois modos de capacidade a seguir.

- Provisionado: escolha esse modo se você conhecer os requisitos de capacidade do seu conector. Você especifica dois valores:
  - O número de operadores.
  - O número de MCUs por trabalhador.
- Escalonamento automático: escolha esse modo se os requisitos de capacidade do seu conector forem variáveis ou se você não os conhecer com antecedência. Quando você usa o modo de

escalabilidade automática, o Amazon MSK Connect substitui a propriedade `tasks.max` do seu conector por um valor proporcional ao número de trabalhadores em execução no conector e ao número de trabalhadores por trabalhador. MCUs

Você especifica três conjuntos de valores:

- O número mínimo e máximo de operadores.
- Os percentuais de expansão e de redução da utilização da CPU, que são determinados pela métrica. `CpuUtilization` Quando a métrica `CpuUtilization` do conector excede o percentual de expansão, o MSK Connect aumenta o número de operadores em execução no conector. Quando a métrica `CpuUtilization` fica abaixo do percentual de expansão, o MSK Connect diminui o número de operadores. O número de operadores sempre permanece dentro dos números mínimo e máximo que você especifica ao criar o conector.
- O número de MCUs por trabalhador.

Para obter mais informações sobre operadores, consulte the section called “Saiba mais sobre os operadores do MSK Connect”. Para saber mais sobre as métricas do MSK Connect, consulte the section called “Monitoramento do MSK Connect”.

Criar um conector

Este procedimento descreve como criar um conector usando o AWS Management Console.

Criando um conector usando o AWS Management Console

1. Abra o console do Amazon MSK em https://console.aws.amazon.com/msk/.
2. No painel esquerdo, em MSK Connect, escolha Conectores.
3. Escolha Criar conector.
4. Você pode escolher entre usar um plug-in personalizado existente para criar o conector ou criar primeiro um novo plug-in personalizado. Para obter informações sobre plug-ins personalizados e como criá-los, consulte the section called “Criar plug-ins personalizados”. Neste procedimento, vamos supor que você tenha um plug-in personalizado que deseja usar. Na lista de plug-ins personalizados, encontre o que você deseja usar, marque a caixa à esquerda e escolha Próximo.
5. Insira um nome e, se desejar, uma descrição.
6. Escolha o cluster ao qual deseja se conectar.

7. Especifique a configuração do conector. Os parâmetros de configuração que você precisa especificar dependerão do tipo de conector que você deseja criar. No entanto, alguns parâmetros são comuns a todos os conectores, por exemplo, os parâmetros `connector.class` e `tasks.max`. Veja a seguir um exemplo de configuração para o Conector de coletor Confluent para Amazon S3.

```
connector.class=io.confluent.connect.s3.S3SinkConnector
tasks.max=2
topics=my-example-topic
s3.region=us-east-1
s3.bucket.name=amzn-s3-demo-bucket
flush.size=1
storage.class=io.confluent.connect.s3.storage.S3Storage
format.class=io.confluent.connect.s3.format.json.JsonFormat
partitioner.class=io.confluent.connect.storage.partitioner.DefaultPartitioner
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.kafka.connect.storage.StringConverter
schema.compatibility=NONE
```

8. Em seguida, configure a capacidade do conector. Você pode escolher entre dois modos de capacidade: provisionado e escalonado automaticamente. Para obter informações sobre essas duas opções, consulte the section called “Saiba mais sobre a capacidade de conectores”.
9. Escolha a configuração padrão do operador ou uma configuração personalizada do operador. Para obter informações sobre como criar configurações personalizadas de operador, consulte the section called “Saiba mais sobre os operadores do MSK Connect”.
10. Em seguida, você especifica o perfil de execução do serviço. Essa deve ser uma função do IAM que o MSK Connect possa assumir e que conceda ao conector todas as permissões necessárias para acessar os AWS recursos necessários. Essas permissões dependem da lógica do conector. Para obter informações sobre como criar essa função, consulte the section called “Saiba mais sobre o perfil de execução do serviço”.
11. Escolha Próximo, revise as informações de segurança e escolha Próximo novamente.
12. Especifique as opções de registro em log que deseja e escolha Próximo. Para obter informações sobre registro em log, consulte the section called “Registro em log”.
13. Escolha Criar conector.

Para usar a API MSK Connect para criar um conector, consulte CreateConnector.

Você pode usar a `UpdateConnector` API para modificar a configuração do conector. Para obter mais informações, consulte the section called “Atualizar um conector”.

Atualizar um conector

Este procedimento descreve como atualizar a configuração de um conector MSK Connect existente usando o. AWS Management Console

Atualizando a configuração do conector usando o AWS Management Console

1. Abra o console do Amazon MSK em https://console.aws.amazon.com/msk/.
2. No painel esquerdo, em MSK Connect, escolha Conectores.
3. Selecione um conector existente.
4. Escolha Editar configuração do conector.
5. Atualize a configuração do conector. Você não pode substituir o `connector.class` uso UpdateConnector de. O exemplo a seguir mostra um exemplo de configuração para o conector Confluent Amazon S3 Sink.

```
connector.class=io.confluent.connect.s3.S3SinkConnector
tasks.max=2
topics=my-example-topic
s3.region=us-east-1
s3.bucket.name=amzn-s3-demo-bucket
flush.size=1
storage.class=io.confluent.connect.s3.storage.S3Storage
format.class=io.confluent.connect.s3.format.json.JsonFormat
partitioner.class=io.confluent.connect.storage.partitioner.DefaultPartitioner
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.kafka.connect.storage.StringConverter
schema.compatibility=NONE
```

6. Selecione Enviar.
7. Em seguida, você pode monitorar o estado atual da operação na guia Operações do conector.

Para usar a API MSK Connect para atualizar a configuração de um conector, consulte UpdateConnector.

Conexão de conectores

As práticas recomendadas a seguir podem melhorar o desempenho da sua conectividade com o Amazon MSK Connect.

Não se sobreponha IPs ao Amazon VPC peering ou ao Transit Gateway

Se você estiver usando o Amazon VPC peering ou o Transit Gateway com o Amazon MSK Connect, não configure seu conector para alcançar os recursos de VPC emparelhados dentro dos intervalos CIDR: IPs

- “10.99.0.0/16”
- “192.168.0.0/16”
- “172.21.0.0/16”

Criar plug-ins personalizados

Um plug-in é um AWS recurso que contém o código que define a lógica do conector. Você carrega um arquivo JAR (ou um arquivo ZIP contendo um ou mais arquivos JAR) em um bucket do S3 e especifica a localização do bucket ao criar o plug-in. Ao criar um conector, você especifica o plug-in que deseja que o MSK Connect use para ele. A relação dos plug-ins com os conectores é one-to-many: Você pode criar um ou mais conectores do mesmo plug-in.

Para obter informações sobre como desenvolver o código para um conector, consulte o Guia de desenvolvimento de conectores na documentação do Apache Kafka.

Criando um plug-in personalizado usando o AWS Management Console

1. Abra o console do Amazon MSK em https://console.aws.amazon.com/msk/.
2. No painel esquerdo, em MSK Connect e escolha Plug-ins personalizados.
3. Escolha Criar plug-in personalizado.
4. Selecione Browse S3 (Navegar no S3).
5. Na lista de buckets do S3, escolha o bucket que contém o arquivo JAR ou ZIP do plug-in.
6. Na lista de objetos, marque a caixa à esquerda do arquivo JAR ou ZIP do plug-in e clique em Escolher.
7. Escolha Criar plug-in personalizado.

Para usar a API MSK Connect para criar um plug-in personalizado, consulte CreateCustomPlugin.

Saiba mais sobre os operadores do MSK Connect

Um operador é um processo de máquina virtual Java (JVM) que executa a lógica do conector. Cada operador cria um conjunto de tarefas que são executadas em threads paralelos e fazem o trabalho de copiar os dados. As tarefas não armazenam o estado e, portanto, podem ser iniciadas, interrompidas ou reiniciadas a qualquer momento para fornecer um pipeline de dados resiliente e escalável. Alterações no número de operadores, seja devido a um evento de escalonamento ou devido a falhas inesperadas, são detectadas automaticamente pelos demais operadores. Eles se organizam para reequilibrar as tarefas no conjunto de operadores restantes. Os operadores do Connect usam os grupos de consumidores do Apache Kafka para coordenar e reequilibrar.

Se os requisitos de capacidade do seu conector forem variáveis ou difíceis de estimar, você pode deixar o MSK Connect escalar o número de operadores conforme necessário entre um limite inferior e um limite superior que você determina. Como alternativa, você pode especificar o número exato de operadores que deseja executar em sua lógica de conector. Para obter mais informações, consulte the section called “Saiba mais sobre a capacidade de conectores”.

Os operadores do MSK Connect consomem endereços IP

Os operadores do MSK Connect consomem endereços IP nas sub-redes fornecidas pelo cliente. Cada operador usa um endereço IP de uma das sub-redes fornecidas pelo cliente. Você deve garantir que tenha endereços IP disponíveis suficientes nas sub-redes fornecidas para uma CreateConnector solicitação para contabilizar uma capacidade especificada, especialmente ao escalar automaticamente conectores em que o número de operadores pode flutuar.

Configuração padrão de operador

O MSK Connect fornece a seguinte configuração padrão de operador:

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

Propriedades de configuração de operador compatíveis

O MSK Connect fornece uma configuração padrão de operador. Também há a opção de criar uma configuração personalizada de operador para usar com os conectores. A lista a seguir inclui

informações sobre as propriedades de configuração do operador compatíveis ou não com o Amazon MSK Connect.

- As propriedades `key.converter` e `value.converter` são obrigatórias.
- O MSK Connect é compatível com as seguintes propriedades de configuração de `producer.`.

```
producer.acks
producer.batch.size
producer.buffer.memory
producer.compression.type
producer.enable.idempotence
producer.key.serializer
producer.linger.ms
producer.max.request.size
producer.metadata.max.age.ms
producer.metadata.max.idle.ms
producer.partitioner.class
producer.reconnect.backoff.max.ms
producer.reconnect.backoff.ms
producer.request.timeout.ms
producer.retry.backoff.ms
producer.value.serializer
```

- O MSK Connect é compatível com as seguintes propriedades de configuração 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
```

- Todas as outras propriedades de configuração que não comecem com os prefixos `producer.` ou `consumer.` são compatíveis, exceto as propriedades a seguir.

```
access.control.
admin.
admin.listeners.https.
client.
connect.
inter.worker.
internal.
listeners.https.
metrics.
metrics.context.
rest.
sasl.
security.
socket.
ssl.
topic.tracking.
worker.
bootstrap.servers
config.storage.topic
connections.max.idle.ms
connector.client.config.override.policy
group.id
listeners
metric.reporters
plugin.path
receive.buffer.bytes
response.http.headers.config
scheduled.rebalance.max.delay.ms
send.buffer.bytes
status.storage.topic
```

Para obter mais informações sobre as propriedades de configuração do operador e o que elas representam, consulte Configurações do Kafka para o Connect na documentação do Apache Kafka.

Criar uma configuração personalizada de operador

Este procedimento descreve como criar uma configuração personalizada de operador usando o AWS Management Console.

Criação de uma configuração personalizada de operador usando o AWS Management Console

1. Abra o console do Amazon MSK em https://console.aws.amazon.com/msk/.
2. No painel esquerdo, em MSK Connect, escolha Configurações do operador.
3. Escolha Criar configuração de operador.
4. Insira um nome e uma descrição opcional e, em seguida, adicione as propriedades e os valores para os quais você deseja defini-los.
5. Escolha Criar configuração de operador.

Para usar a API do MSK Connect para criar uma configuração de operador, consulte CreateWorkerConfiguration.

Gerenciar deslocamentos do conector de origem usando `offset.storage.topic`

Esta seção fornece informações para ajudar você a gerenciar os deslocamentos do conector de origem usando o tópico de deslocamento de armazenamento. O tópico de deslocamento de armazenamento é um tópico interno que o Kafka Connect usa para armazenar deslocamentos de configuração de conectores e tarefas.

Considerações

Considere o seguinte ao gerenciar os deslocamentos do conector de origem.

- Para especificar um tópico de deslocamento de armazenamento, forneça o nome do tópico do Kafka no qual os deslocamentos do conector são armazenados como o valor `offset.storage.topic` em sua configuração de operador.
- Tenha cuidado ao fazer alterações na configuração de um conector. A alteração dos valores da configuração pode resultar em um comportamento não intencional do conector se um conector de origem usar valores da configuração para os principais registros de deslocamento. Recomendamos que você consulte a documentação do seu plug-in para obter orientação.

- Personalize o número padrão de partições: além de personalizar a configuração do operador adicionando `offset.storage.topic`, você pode personalizar o número de partições para os tópicos de deslocamento e armazenamento de status. As partições padrão para tópicos internos são as seguintes.
  - `config.storage.topic`: 1, não configurável, deve ser tópico de partição única
  - `offset.storage.topic`: 25, configurável fornecendo `offset.storage.partitions`
  - `status.storage.topic`: 5, configurável fornecendo `status.storage.partitions`
- Exclusão manual de tópicos: o Amazon MSK Connect cria novos tópicos internos do Kafka Connect (o nome do tópico começa com `__amazon_msk_connect`) em cada implantação de conectores. Tópicos antigos anexados a conectores excluídos não são removidos automaticamente porque tópicos internos, como `offset.storage.topic`, podem ser reutilizados entre conectores. No entanto, você pode excluir manualmente tópicos internos não utilizados criados pelo MSK Connect. Os tópicos internos são nomeados segundo o formato `__amazon_msk_connect_<offsets|status|configs>_`*`connector_name_connector_id`*.

  É possível usar a expressão regular `__amazon_msk_connect_<offsets|status|configs>_`*`connector_name_connector_id`* para excluir os tópicos internos. Você não deve excluir um tópico interno que esteja sendo usado atualmente por um conector em execução.
- Usar o mesmo nome para os tópicos internos criados pelo MSK Connect: se quiser reutilizar o tópico de deslocamento de armazenamento para consumir deslocamentos de um conector criado anteriormente, você deverá dar ao novo conector o mesmo nome do conector antigo. A propriedade `offset.storage.topic` pode ser definida usando a configuração do operador para atribuir o mesmo nome ao `offset.storage.topic` e reutilizada entre conectores diferentes. Essa configuração é descrita em Gerenciamento de deslocamentos de conectores. O MSK Connect não permite que conectores diferentes compartilhem `config.storage.topic` e `status.storage.topic`. Esses tópicos são criados sempre que você cria um novo conector no MSKC. Eles são nomeados automaticamente de acordo com o formato `__amazon_msk_connect_<status|configs>_`*`connector_name_connector_id`* e, portanto, são diferentes nos diferentes conectores que você cria.

Usar o tópico padrão de deslocamento de armazenamento

Por padrão, o Amazon MSK Connect gera um novo tópico de deslocamento de armazenamento em seu cluster do Kafka para cada conector que você cria. O MSK estrutura o nome do tópico padrão

usando partes do ARN do conector. Por exemplo, `.__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-4abc-8be8-c657f7e4ff32-2`

Usar um tópico personalizado de deslocamento de armazenamento

Para fornecer continuidade de deslocamento entre conectores de origem, você pode usar um tópico de deslocamento de armazenamento de sua escolha em vez do tópico padrão. Especificar um tópico de deslocamento de armazenamento ajuda você a realizar tarefas como criar um conector de origem que retoma a leitura desde o último deslocamento de um conector anterior.

Para especificar um tópico de deslocamento de armazenamento, você fornece um valor para a propriedade `offset.storage.topic` em sua configuração de operador antes de criar um conector. Se quiser reutilizar o tópico de deslocamento de armazenamento para consumir deslocamentos de um conector criado anteriormente, você deverá dar ao novo conector o mesmo nome do conector antigo. Se você criar um tópico personalizado de deslocamento de armazenamento, deverá definir `cleanup.policy` como `compact` na configuração do tópico.

> **Note**
>
> Se você especificar um tópico de deslocamento de armazenamento ao criar um conector de coletor, o MSK Connect criará o tópico se ele ainda não existir. No entanto, o tópico não será usado para armazenar deslocamentos de conectores.
> Em vez disso, os deslocamentos do conector do coletor serão gerenciados usando o protocolo de grupo de consumidores Kafka. Cada conector de coletor cria um grupo chamado `connect-{CONNECTOR_NAME}`. Enquanto o grupo de consumidores existir, todos os conectores de coletor sucessivos que você criar com o mesmo valor `CONNECTOR_NAME` continuarão a partir do último deslocamento confirmado.

Example Especificar um tópico de deslocamento de armazenamento para recriar um conector de origem com uma configuração atualizada

Suponha que você tenha um conector de Change Data Capture (CDC – Captura de dados de alteração) e queira modificar a configuração do conector sem perder seu lugar no fluxo do CDC. Não é possível atualizar a configuração do conector existente, mas você pode excluir o conector e criar um novo com o mesmo nome. Para informar ao novo conector por onde começar a leitura no fluxo do CDC, você pode especificar o tópico de deslocamento de armazenamento do conector antigo em sua configuração de operador. As etapas a seguir demonstram como concluir essa tarefa.

1. Em sua máquina cliente, execute o comando a seguir para encontrar o nome do tópico de deslocamento de armazenamento do seu conector. Substitua *<bootstrapBrokerString>* pela string do agente de bootstrap do seu cluster. Para obter instruções sobre como obter sua string de agente de bootstrap, consulte Obter os agentes de bootstrap para um cluster do Amazon MSK.

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

   A saída a seguir mostra uma lista de todos os tópicos do cluster, incluindo qualquer tópico de conector interno padrão. Neste exemplo, o conector CDC existente usa o tópico padrão de deslocamento de armazenamento criado pelo MSK Connect. É por isso que o tópico de deslocamento de armazenamento é chamado de `__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 o console do Amazon MSK em https://console.aws.amazon.com/msk/.
3. Escolha seu conector na lista Conectores. Copie e salve o conteúdo do campo Configuração do conector para que você possa modificá-lo e usá-lo na criação do novo conector.
4. Selecione Excluir para excluir o conector. Em seguida, insira o nome do conector no campo de entrada de texto para confirmar a exclusão.
5. Crie uma configuração personalizada de operador com valores adequados ao seu cenário. Para instruções, consulte Criar uma configuração personalizada de operador.

   Em sua configuração de operador, você deve especificar o nome do tópico de deslocamento de armazenamento que você recuperou anteriormente como o valor de `offset.storage.topic`, assim como na configuração a seguir.

   ```
   config.providers.secretManager.param.aws.region=eu-west-3
   ```

```
key.converter=<org.apache.kafka.connect.storage.StringConverter>
value.converter=<org.apache.kafka.connect.storage.StringConverter>
config.providers.secretManager.class=com.github.jcustenborder.kafka.config.aws.SecretsManag
config.providers=secretManager
offset.storage.topic=__amazon_msk_connect_offsets_my-mskc-
connector_12345678-09e7-4abc-8be8-c657f7e4ff32-2
```

6. **Important**

   Você deve dar ao seu novo conector o mesmo nome do conector antigo.

   Crie um novo conector usando a configuração de operador que você definiu na etapa anterior. Para instruções, consulte Criar um conector.

Tutorial: Externalizar informações confidenciais usando provedores de configuração

Este exemplo mostra como externalizar informações confidenciais para o Amazon MSK Connect usando um provedor de configuração de código aberto. Um provedor de configuração permite que você especifique variáveis, em vez de texto simples, em uma configuração de conector ou de operador, e os operadores em execução em seu conector resolvem essas variáveis em runtime. Isso evita que credenciais e outros segredos sejam armazenados em texto simples. O provedor de configuração no exemplo suporta a recuperação de parâmetros de configuração do AWS Secrets Manager, Amazon S3 e Systems Manager (SSM). Na Etapa 2, você verá como configurar o armazenamento e a recuperação de informações confidenciais para o serviço que deseja configurar.

Considerações

Considere o seguinte ao usar o provedor de configuração do MSK com o Amazon MSK Connect:

- Atribua as permissões adequadas ao usar os provedores de configuração para o perfil de execução de serviços do IAM.
- Defina os provedores de configuração nas configurações de trabalho e sua implementação na configuração do conector.
- Valores confidenciais de configuração podem aparecer nos registros do conector se um plug-in não definir esses valores como secretos. O Kafka Connect trata valores de configuração

indefinidos da mesma forma que qualquer outro valor de texto simples. Para saber mais, consulte Como evitar que segredos apareçam nos logs do conector.

- Por padrão, o MSK Connect reinicia frequentemente um conector quando o conector usa um provedor de configuração. Para desativar esse comportamento de reinicialização, você pode definir o valor `config.action.reload` como `none` na configuração do conector.

Criar um plug-in personalizado e fazer o upload para o S3

Para criar um plug-in personalizado, crie um arquivo zip que contenha o conector e o msk-config-provider executando os seguintes comandos em sua máquina local.

Para criar um plug-in personalizado usando uma janela de terminal e o Debezium como conector

Use a AWS CLI para executar comandos como superusuário com credenciais que permitem acessar seu bucket do S3. AWS Para obter informações sobre como instalar e configurar a AWS CLI, consulte Introdução à AWS CLI no Guia do usuário.AWS Command Line Interface Para obter informações sobre o uso da AWS CLI com o Amazon S3, consulte Usando o Amazon S3 com a AWS CLI no Guia do usuário.AWS Command Line Interface

1. Em uma janela de terminal, crie uma pasta nomeada `custom-plugin` no seu espaço de trabalho usando o comando a seguir.

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

2. Baixe a versão estável mais recente do plug-in MySQL Connector no site do Debezium usando o comando a seguir.

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

   Extraia o arquivo gzip baixado na pasta `custom-plugin` usando o comando a seguir.

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

3. Baixe o arquivo zip do provedor de configuração do MSK usando o comando a seguir.

   ```
   wget https://github.com/aws-samples/msk-config-providers/releases/download/r0.4.0/
   msk-config-providers-0.4.0-with-dependencies.zip
   ```

Extraia o arquivo zip baixado na `custom-plugin` pasta usando o comando a seguir.

```
unzip msk-config-providers-0.4.0-with-dependencies.zip
```

4. Compacte o conteúdo do provedor de configuração do MSK da etapa acima e do conector personalizado em um só arquivo chamado `custom-plugin.zip`.

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

5. Faça upload do arquivo para o S3 para referência posterior.

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

6. No console do Amazon MSK, na seção MSK Connect, escolha Custom Plugin, depois escolha Create custom plugin e navegue pelo bucket s3: **< *S3_URI_BUCKET_LOCATION* >** S3 para selecionar o arquivo ZIP do plug-in personalizado que você acabou de enviar.

7. Insira **`debezium-custom-plugin`** para o nome do plug-in. Opcionalmente, insira uma descrição e escolha Criar um plug-in personalizado.

Configurar parâmetros e permissões para diferentes provedores

Você pode configurar valores de parâmetros nestes três serviços:

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

Escolha uma das guias abaixo para obter instruções sobre como configurar parâmetros e permissões relevantes para esse serviço.

Configure in Secrets Manager

Para configurar valores de parâmetros no Secrets Manager

1. Abra o console do Secrets Manager.

2. Crie um novo segredo para armazenar suas credenciais ou segredos. Para obter instruções, consulte Criar um AWS Secrets Manager segredo no Guia AWS Secrets Manager do usuário.
3. Copie o ARN do seu segredo.
4. Adicione as permissões do Secrets Manager do exemplo de política a seguir ao seu perfil de execução de serviço. Substitua o ARN de exemplo,`arn:aws:secretsmanager:us-east-1:123456789012:secret:MySecret-1234`, pelo ARN do seu segredo.
5. Adicione a configuração do operador e as instruções do 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 usar o provedor de configuração do Secrets Manager, copie as seguintes linhas de código na caixa de texto de configuração do operador na Etapa 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 o provedor de configuração do Secrets Manager, copie as seguintes linhas de código na configuração do conector na Etapa 4.

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

Você também pode usar a etapa acima com mais provedores de configuração.

Configure in Systems Manager Parameter Store

Para configurar valores de parâmetros no Systems Manager Parameter Store

1. Abra o console do Systems Manager.
2. No painel de navegação, selecione Parameter Store (Repositório de parâmetros).
3. Crie um novo parâmetro para armazenar no Systems Manager. Para obter instruções, consulte Criar um parâmetro do Systems Manager (console) no Guia AWS Systems Manager do usuário.
4. Copie o ARN do seu parâmetro.
5. Adicione as permissões do Systems Manager do exemplo de política a seguir ao seu perfil de execução de serviço. *<arn:aws:ssm:us-east-1:123456789000:parameter/MyParameterName>*Substitua pelo ARN do seu parâmetro.

```
{
        "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 usar o provedor de configuração do Parameter Store, copie as seguintes linhas de código na caixa de texto de configuração do operador na Etapa 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 o provedor de configuração do Parameter Store, copie as seguintes linhas de código na configuração do conector na Etapa 5.

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

Você também pode agrupar as duas etapas acima com mais provedores de configuração.

Configure in Amazon S3

Para configurar objects/files no Amazon S3

1. Abra o console Amazon S3.
2. Carregue um objeto para um bucket no S3. Para obter instruções, consulte Carregar objetos.
3. Copie o ARN do seu objeto.

4. Adicione as permissões de leitura de objeto do Amazon S3 do exemplo de política a seguir ao seu perfil de execução de serviço. Substitua o exemplo ARN,`arn:aws:s3:::`*<MY_S3_BUCKET/path/to/custom-plugin.zip>*, pelo ARN do seu 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 usar o provedor de configuração do Amazon S3, copie as seguintes linhas de código na caixa de texto de configuração do operador na Etapa 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 o provedor de configuração do Amazon S3, copie as seguintes linhas de código na configuração do conector na Etapa 4.

```
#Example implementation for S3 object

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

Você também pode agrupar as duas etapas acima com mais provedores de configuração.

Criar uma configuração personalizada de operador com informações sobre seu provedor de configuração

1. Selecione as Configurações do operador na seção Amazon MSK Connect.
2. Selecione Criar configuração de operador.
3. Digite `SourceDebeziumCustomConfig` na caixa de texto Nome da configuração do operador. A descrição é opcional.
4. Copie o código de configuração relevante com base nos provedores desejados e cole-o na caixa de texto de Configuração do operador.
5. Este é um exemplo da configuração de operador para todos os três provedores:

```
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. Clique em Criar configuração de operador.

Criar o conector

1. Crie um novo conector usando as instruções em Criar um novo conector.
2. Escolha o arquivo `custom-plugin.zip` que você enviou para o bucket do S3 em ??? como origem do plug-in personalizado.
3. Copie o código de configuração relevante com base nos provedores desejados e cole-o no campo Configuração do conector.
4. Este é um exemplo da configuração do conector para todos os três provedores:

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

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

#Example implementation for Amazon S3 file/object
database.ssl.truststore.location = ${s3import:us-west-2:my_cert_bucket/path/to/
trustore_unique_filename.jks}
```

5. Selecione Usar uma configuração personalizada e escolha SourceDebeziumCustomConfigno menu suspenso Configuração do trabalhador.
6. Siga as etapas restantes das instruções em Criar conector.

Perfis e políticas do IAM para o MSK Connect

Esta seção ajuda você a configurar as políticas e funções apropriadas do IAM para implantar e gerenciar com segurança o Amazon MSK Connect em AWS seu ambiente. As seções a seguir explicam a função de execução do serviço que deve ser usada com o MSK Connect, incluindo a política de confiança necessária e as permissões adicionais necessárias ao se conectar a um cluster MSK autenticado pelo IAM. A página também fornece exemplos de políticas abrangentes do IAM para conceder acesso total à funcionalidade do MSK Connect, bem como detalhes sobre as políticas AWS gerenciadas disponíveis para o serviço.

Tópicos

- Saiba mais sobre o perfil de execução do serviço
- Exemplo de política do IAM para o MSK Connect

- Prevenção do problema confused deputy entre serviços
- AWS políticas gerenciadas para o MSK Connect
- Usar perfis vinculados ao serviço para o MSK Connect

Saiba mais sobre o perfil de execução do serviço

> **Note**
>
> O Amazon MSK Connect não é compatível com o uso do perfil vinculado a serviço como o perfil de execução do serviço. É necessário criar um perfil de execução do serviço distinto. Para obter instruções sobre como criar uma função personalizada do IAM, consulte Como criar uma função para delegar permissões a um AWS serviço no Guia do usuário do IAM.

Ao criar um conector com o MSK Connect, você precisa especificar uma função AWS Identity and Access Management (IAM) para usar com ele. Seu perfil de execução do serviço deve ter a seguinte política de confiança para que o MSK Connect possa assumi-lo. Para obter informações sobre as chaves de contexto de condição, consulte the section called “Prevenção do problema confused deputy entre serviços”.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "kafkaconnect.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:kafkaconnect:us-
east-1:123456789012:connector/myConnector/abc12345-abcd-4444-a8b9-123456f513ed-2"
        }
```

```
      }
    }
  ]
}
```

Se o cluster Amazon MSK que você deseja usar com seu conector for um cluster que usa autenticação do IAM, será necessário adicionar a seguinte política de permissões ao perfil de execução do serviço do conector. Para obter informações sobre como encontrar o UUID do seu cluster e como criar um tópico ARNs, consulte. the section called “Recursos da política de autorização”

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-*"
            ]
        }
    ]
}
```

Dependendo do tipo de conector, talvez você também precise anexar à função de execução do serviço uma política de permissões que permita que ela acesse AWS recursos. Por exemplo,

se seu conector precisar enviar dados para um bucket do S3, o perfil de execução do serviço deverá ter uma política de permissões que conceda permissão para gravar nesse bucket. Para fins de teste, você pode usar uma das políticas predefinidas do IAM que dão acesso total, como `arn:aws:iam::aws:policy/AmazonS3FullAccess`. No entanto, por motivos de segurança, recomendamos que você use a política mais restritiva que permita que seu conector leia da AWS fonte ou grave no AWS coletor.

Exemplo de política do IAM para o MSK Connect

Para fornecer acesso total a todas as funcionalidades do MSK Connect a um usuário não administrador, anexe uma política como a seguinte ao perfil do IAM do usuário.

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/*"
      ]
    }
  ]
}
```

Prevenção do problema confused deputy entre serviços

"Confused deputy" é um problema de segurança no qual uma entidade sem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Em AWS, a falsificação de identidade entre serviços pode resultar no problema confuso do deputado. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, a AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta.

Recomendamos usar as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` em políticas de recursos para limitar as permissões que o MSK Connect concede a outro serviço para o recurso. Se o valor `aws:SourceArn` não contiver o ID da conta (p. ex., um ARN de um bucket do Amazon S3 não contiver o ID da conta), você deverá usar ambas as chaves de contexto de condição global para limitar as permissões. Se você utilizar ambas as chaves de contexto de condição global e o valor de `aws:SourceArn` contiver o ID da conta, o valor de `aws:SourceAccount` e a conta no valor de `aws:SourceArn` deverão utilizar o mesmo ID de conta quando utilizados na mesma declaração da política. Use `aws:SourceArn` se quiser apenas um recurso associado a acessibilidade de serviço. Use `aws:SourceAccount` se quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços.

No caso do MSK Connect, o valor de `aws:SourceArn` deve ser um conector do MSK.

A maneira mais eficaz de se proteger do problema 'confused deputy' é usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso. Se você não souber o ARN completo do recurso ou se estiver especificando vários recursos, use a chave de condição de contexto global `aws:SourceArn` com curingas (*) para as partes desconhecidas do ARN. Por exemplo, *`arn:aws:kafkaconnect:us-east-1:123456789012:connector/*`* representa todos os conectores que pertencem à conta com ID 123456789012 na região Leste dos EUA (Norte da Virgínia).

O exemplo a seguir mostra como é possível usar as chaves de contexto de condição globais `aws:SourceArn` e `aws:SourceAccount` no MSK Connect para evitar o problema "confused deputy". Substitua *`123456789012`* e arn:aws:kafkaconnect: ::connector//pelas informações suas e do conector*`us-east-1`*. *`123456789012`* *`my-S3-Sink-Connector`* *`abcd1234-5678-90ab-cdef-1234567890ab`* Conta da 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 gerenciadas para o MSK Connect

Uma política AWS gerenciada é uma política autônoma criada e administrada por AWS. AWS as políticas gerenciadas são projetadas para fornecer permissões para muitos casos de uso comuns, para que você possa começar a atribuir permissões a usuários, grupos e funções.

Lembre-se de que as políticas AWS gerenciadas podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque elas estão disponíveis para uso de todos os AWS clientes. Recomendamos que você reduza ainda mais as permissões definindo as políticas gerenciadas pelo cliente que são específicas para seus casos de uso.

Você não pode alterar as permissões definidas nas políticas AWS gerenciadas. Se AWS atualizar as permissões definidas em uma política AWS gerenciada, a atualização afetará todas as identidades principais (usuários, grupos e funções) às quais a política está anexada. AWS é mais provável que atualize uma política AWS gerenciada quando uma nova AWS service (Serviço da AWS) for lançada ou novas operações de API forem disponibilizadas para serviços existentes.

Para mais informações, consulte Políticas gerenciadas pela AWS no Manual do usuário do IAM.

AWS política gerenciada: Amazon MSKConnect ReadOnlyAccess

Essa política concede ao usuário as permissões necessárias para listar e descrever os recursos do MSK Connect.

É possível anexar a política `AmazonMSKConnectReadOnlyAccess` às identidades do 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 gerenciada: KafkaConnectServiceRolePolicy

Essa política concede ao serviço MSK Connect as permissões necessárias para criar e gerenciar interfaces de rede que tenham a tag `AmazonMSKConnectManaged:true`. Essas interfaces de rede permitem que a rede do MSK Connect acesse os recursos em sua Amazon VPC, como um cluster do Apache Kafka ou uma origem ou um coletor.

Você não pode se vincular KafkaConnectServiceRolePolicy às suas entidades do IAM. Essa política é anexada a um perfil vinculado a serviço que permite que o MSK Connect realize ações em seu nome.

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"
     }
    }
   }
  ]
}
```

Atualizações do MSK Connect para políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas do MSK Connect desde que esse serviço começou a rastrear essas alterações.

Alteração	Descrição	Data
Atualização da política somente leitura do MSK Connect	O MSK Connect atualizou a MSKConnect ReadOnlyA ccess política da Amazon para remover as restrições nas operações de listagem.	13 de outubro de 2021
O MSK Connect começou a monitorar alterações	O MSK Connect começou a monitorar as mudanças em suas políticas AWS gerenciad as.	14 de setembro de 2021

Usar perfis vinculados ao serviço para o MSK Connect

O Amazon MSK Connect usa funções AWS Identity and Access Management vinculadas a serviços (IAM). Um perfil vinculado a serviço é um tipo especial de perfil do IAM vinculado diretamente ao MSK Connect. As funções vinculadas ao serviço são predefinidas pelo MSK Connect e incluem todas as permissões que o serviço exige para chamar outros AWS serviços em seu nome.

Um perfil vinculado a serviço facilita a configuração do MSK Connect porque você não precisa adicionar as permissões necessárias manualmente. O MSK Connect define as permissões dos perfis vinculados ao serviço e, exceto se definido de outra forma, somente o MSK Connect pode assumir seus perfis. As permissões definidas incluem a política de confiança e a política de permissões, que não pode ser anexada a nenhuma outra entidade do IAM.

Para obter informações sobre outros serviços compatíveis com funções vinculadas a serviços, consulte Serviços da AWS compatíveis com o IAM e procure os serviços que contenham Sim na coluna Função vinculada ao serviço. Escolha um Sim com um link para visualizar a documentação do perfil vinculado para esse serviço.

Permissões de perfil vinculado a serviço para o MSK Connect

O MSK Connect usa a função vinculada ao serviço chamada — AWSServiceRoleForKafkaConnectPermite que o Amazon MSK Connect acesse os recursos da Amazon em seu nome.

A função AWSService RoleForKafkaConnect vinculada ao serviço confia no `kafkaconnect.amazonaws.com` serviço para assumir a função.

Para obter mais informações sobre a política de permissões usada pelo perfil, consulte the section called “KafkaConnectServiceRolePolicy”.

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua um perfil vinculado a serviço. Para mais informações, consulte Permissões de perfil vinculado ao serviço no Guia do usuário do IAM.

Criação de um perfil vinculado a serviço para o MSK Connect

Não é necessário criar manualmente um perfil vinculado ao serviço. Quando você cria um conector na AWS Management Console, na ou na AWS API AWS CLI, o MSK Connect cria a função vinculada ao serviço para você.

Se excluir esse perfil vinculado ao serviço e precisar criá-lo novamente, será possível usar esse mesmo processo para recriar o perfil em sua conta. Quando você cria um conector, o MSK Connect cria um perfil vinculado a serviço para você novamente.

Edição de um perfil vinculado a serviço para o MSK Connect

O MSK Connect não permite que você edite a função vinculada ao AWSService RoleForKafkaConnect serviço. Depois que você criar um perfil vinculado ao serviço, não poderá alterar o nome do perfil, pois várias entidades podem fazer referência ao perfil. No entanto, você poderá editar a descrição do perfil usando o IAM. Para obter mais informações, consulte Editar uma função vinculada a serviço no Guia do usuário do IAM.

Exclusão de um perfil vinculado a serviço para o MSK Connect

Você pode usar o console do IAM AWS CLI ou a AWS API para excluir manualmente a função vinculada ao serviço. Para isso, primeiro é necessário excluir manualmente todos os conectores do MSK Connect e excluir o perfil manualmente. Para obter mais informações, consulte Excluir um perfil vinculado ao serviço no Guia do usuário do IAM.

Regiões compatíveis com perfis vinculados a serviço do MSK Connect

O MSK Connect é compatível com perfis vinculados a serviço em todas as regiões nas quais o serviço esteja disponível. Para mais informações, consulte Regiões e endpoints da AWS.

Habilitar o acesso à internet para o Amazon MSK Connect

Se o seu conector para o Amazon MSK Connect precisar de acesso à Internet, recomendamos que você use as seguintes configurações Amazon Virtual Private Cloud (VPC) para habilitar esse acesso.

- Configure seu conector com sub-redes privadas.
- Crie um gateway NAT público ou uma instância NAT pública para sua VPC em uma sub-rede pública. Para obter mais informações, consulte a página Conectar sub-redes à Internet ou a outros dispositivos VPCs usando NAT no Guia do Amazon Virtual Private Cloudusuário.
- Permita o tráfego de saída de suas sub-redes privadas para seu gateway ou instância NAT.

Configurar um gateway NAT para o Amazon MSK Connect

As etapas a seguir mostram como configurar um gateway NAT para permitir o acesso à Internet para um conector. Você deve concluir estas etapas antes de criar um conector em uma sub-rede privada.

Concluir pré-requisitos para configurar um gateway NAT

Certifique-se de ter os seguintes itens.

- O ID do Amazon Virtual Private Cloud (VPC) associado ao seu cluster. Por exemplo, vpc-123456ab.
- A IDs das sub-redes privadas em sua VPC. Por exemplo, subnet-a1b2c3de, subnet-f4g5h6ij etc. Você deve configurar seu conector com sub-redes privadas.

Habilitar o acesso à internet para o conector

Para habilitar o acesso à Internet para seu conector

1. Abra o Amazon Virtual Private Cloud console em https://console.aws.amazon.com/vpc/.
2. Crie uma sub-rede pública para seu gateway NAT com um nome descritivo e anote o ID da sub-rede. Para obter instruções detalhadas, consulte Criar uma sub-rede na VPC.
3. Crie um gateway da Internet para que a VPC possa se comunicar com a Internet e anote o ID do gateway. Anexe o gateway da internet à sua VPC. Para obter mais instruções, consulte Criar e anexar um gateway da Internet à VPC.
4. Provisione um gateway NAT público para que os hosts em suas sub-redes privadas possam acessar sua sub-rede pública. Ao criar o gateway NAT, selecione a sub-rede pública que você criou anteriormente. Para obter instruções, consulte Create a NAT gateway (Criar um gateway NAT)
5. Configure suas tabelas de rotas. Para concluir essa configuração, você deve ter duas tabelas de rotas no total. Você já deve ter uma tabela de rotas principal criada automaticamente junto com sua VPC. Nesta etapa, você cria uma tabela de rotas adicional para sua sub-rede pública.

    a. Use as configurações a seguir para modificar a tabela de rotas principal da sua VPC para que suas sub-redes privadas roteiem o tráfego para seu gateway NAT. Para obter instruções, consulte Trabalhar com tabelas de rotas no Guia do usuário do Amazon Virtual Private Cloud.

Tabela de rotas MSKC privado

Propriedade	Valor
Name tag	Recomendamos que você atribua uma tag de nome descritivo a essa tabela de rotas para ajudar na identificação dela. Por exemplo, MSKC privado.
Sub-redes associadas	Suas sub-redes privadas
Uma rota para habilitar o acesso à Internet para o MSK Connect	• Destino: 0.0.0.0/0 • Alvo: o ID do seu gateway NAT Por exemplo, nat-12a345bc6789efg1h.
Uma rota local para o tráfego interno	• Destino: 10.0.0.0/16 Esse valor pode ser diferente dependendo do bloco CIDR da sua VPC. • Alvo: local

b. Siga as instruções em Criar uma tabela de rotas personalizada para criar uma tabela de rotas para sua sub-rede pública. Ao criar a tabela, insira um nome descritivo no campo Tag de nome para ajudar você a identificar a qual sub-rede a tabela está associada. Por exemplo, MSKC público.

c. Configure sua tabela de rotas MSKC público usando as configurações a seguir.

Propriedade	Valor
Name tag	MSKC público ou um nome descritivo diferente que você escolher
Sub-redes associadas	Sua sub-rede pública com gateway NAT
Uma rota para habilitar o acesso à Internet para o MSK Connect	• Destino: 0.0.0.0/0 • Alvo: o ID do seu gateway da Internet Por exemplo, igw-1a234bc5.

Propriedade	Valor
Uma rota local para o tráfego interno	• Destino: 10.0.0.0/16 Esse valor pode ser diferente dependendo do bloco CIDR da sua VPC. • Alvo: local

Saiba mais sobre nomes de host DNS privados

Com o suporte a nomes de host DNS privados no MSK Connect, você pode configurar conectores para consultar nomes de domínio públicos ou privados. O suporte dependerá dos servidores DNS especificados no Conjunto de opções de DHCP da VPC.

Um conjunto de opções de DHCP é um grupo de configurações de rede que EC2 instâncias usam em uma VPC para comunicação pela rede da VPC. Cada VPC tem um conjunto padrão de opções de DHCP, mas você pode criar um conjunto personalizado de opções de DHCP se quiser que as instâncias em sua VPC usem um servidor de DNS diferente para a resolução de nomes de domínio em vez do servidor DNS fornecido pela Amazon. Consulte Conjuntos de opções de DHCP na Amazon VPC.

Antes da inclusão da capacidade/recurso de resolução de DNS privado no MSK Connect, os conectores usavam o serviço de resolvedores de DNS da VPC para consultas de DNS de um conector do cliente. Os conectores não usavam os servidores DNS definidos nos conjuntos de opções de DHCP da VPC do cliente para a resolução de DNS.

Os conectores só podiam consultar nomes de host nas configurações de conectores do cliente ou em plug-ins que fossem resolvíveis publicamente. Eles não conseguiam resolver nomes de host privados definidos em uma zona hospedada de maneira privada nem usar servidores DNS em outra rede de clientes.

Sem o DNS privado, os clientes que optaram por tornar seus bancos de dados, data warehouses e sistemas como o Secrets Manager em sua própria VPC inacessíveis à Internet não poderiam trabalhar com conectores do MSK. Geralmente os clientes usam nomes de host DNS privados para atender à postura de segurança corporativa.

Configurar um conjunto de opções de DHCP da VPC para o conector

Os conectores usam automaticamente os servidores DNS definidos em seu conjunto de opções de DHCP da VPC quando o conector é criado. Antes de criar um conector, certifique-se de configurar o conjunto de opções de DHCP da VPC para os requisitos de resolução de nome de host DNS do seu conector.

Os conectores criados antes da disponibilização do recurso de nome de host DNS privado no MSK Connect continuam usando a configuração de resolução de DNS anterior sem necessidade de modificação.

Se você precisar apenas de uma resolução de nome de host DNS que possa ser resolvida publicamente em seu conector, para facilitar a configuração, recomendamos usar a VPC padrão da sua conta ao criar o conector. Consulte o Servidor DNS da Amazon no Guia do usuário da Amazon VPC para obter mais informações sobre o servidor DNS fornecido pela Amazon ou sobre o Amazon Route 53 Resolver.

Se você precisar resolver nomes de host DNS privados, certifique-se de que a VPC transmitida durante a criação do conector tenha suas opções de DHCP configuradas corretamente. Para obter mais informações, consulte Trabalhar com conjuntos de opções de DHCP no Guia do usuário da Amazon VPC.

Ao configurar um conjunto de opções de DHCP para resolução de nome de host DNS privado, certifique-se de que o conector possa acessar os servidores DNS personalizados que você configurar no conjunto de opções de DHCP. Caso contrário, a criação do conector falhará.

Após personalizar o conjunto de opções de DHCP da VPC, os conectores criados posteriormente nessa VPC usarão os servidores DNS que você especificou no conjunto de opções. Se você alterar o conjunto de opções após criar um conector, o conector adotará as configurações do novo conjunto de opções em alguns minutos.

Configurar atributos de DNS para a VPC

Certifique-se de ter os atributos de DNS da VPC configurados corretamente conforme descrito em Atributos de DNS em sua VPC e Nomes de host DNS no Guia do usuário da Amazon VPC.

Consulte Como resolver consultas de DNS entre VPCs e sua rede no Guia do desenvolvedor do Amazon Route 53 para obter informações sobre o uso de endpoints de resolução de entrada e de saída para conectar outras redes à sua VPC e trabalhar com seu conector.

Resolver falhas na criação do conector

Esta seção descreve possíveis falhas na criação de conectores associadas à resolução de DNS e ações sugeridas para resolver os problemas.

Falha	Ação sugerida
A criação do conector falhará se uma consulta de resolução de DNS falhar ou se os servidores DNS estiverem inacessíveis pelo conector.	Você poderá observar falhas na criação de conectores devido a consultas infrutíferas de resolução de DNS em seus CloudWatch logs, se tiver configurado esses logs para seu conector. Verifique as configurações do servidor DNS e garanta a conectividade de rede com os servidores DNS pelo conector.
Se você alterar a configuração dos servidores DNS no conjunto de opções de DHCP da VPC enquanto um conector estiver em execução, as consultas de resolução de DNS do conector poderão falhar. Se a resolução de DNS falhar, algumas das tarefas do conector podem entrar em um estado de falha.	Você poderá observar falhas na criação de conectores devido a consultas infrutíferas de resolução de DNS em seus CloudWatch logs, se tiver configurado esses logs para seu conector. As tarefas com falha deverão reiniciar automaticamente para que o conector volte a funcionar. Se isso não acontecer, você pode entrar em contato com o suporte para reiniciar as tarefas que falharam no conector ou recriar o conector.

Segurança no MSK Connect

Você pode usar um endpoint da VPC Connect de interface, desenvolvido por AWS PrivateLink, para impedir que o tráfego entre sua Amazon VPC e o Amazon MSK Connect saia da rede da Amazon. APIs Os endpoints da VPC de interface não exigem um gateway da Internet, dispositivo NAT, conexão VPN ou conexão do. AWS Direct Connect Para obter mais informações, consulte Use o Amazon MSK APIs com endpoints de interface VPC.

Registro em log no MSK Connect

O MSK Connect pode gravar eventos de log que você pode usar para depurar seu conector. Ao criar um conector, você pode especificar zero ou mais dos seguintes destinos de log:

- Amazon CloudWatch Logs: você especifica o grupo de logs para o qual deseja que o MSK Connect envie os eventos de log do seu conector. Para obter informações sobre como criar um grupo de registros, consulte Criar um grupo de registros no Guia do usuário de CloudWatch registros.
- Amazon S3: você especifica o bucket do S3 para o qual deseja que o MSK Connect envie os eventos de log do seu conector. Para obter mais informações sobre como criar um bucket do S3, consulte Criar um bucket, no Guia do usuário do Amazon S3.
- Amazon Data Firehose: você especifica o stream de entrega para o qual deseja que o MSK Connect envie os eventos de log do conector. Para obter informações sobre como criar um stream de entrega, consulte Creating an Amazon Data Firehose delivery stream no Guia do usuário do Firehose.

Para saber mais sobre como configurar o registro em log, consulte Habilitar o registro em log de determinados serviços da AWS no Guia do usuário do Amazon CloudWatch Logs .

O MSK Connect emite os seguintes tipos de eventos de log:

Nível	Descrição
`INFO`	Eventos de runtime de interesse na inicializ ação e no desligamento.
`WARN`	Situações de runtime que não são erros, mas são indesejáveis ou inesperadas.
`FATAL`	Erros graves que causam encerramento prematuro.
`ERROR`	Condições inesperadas e erros de runtime que não são fatais.

Veja a seguir um exemplo de um evento de registro enviado para o 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)
```

Como evitar que segredos apareçam nos logs do conector

> **Note**
>
> Valores confidenciais de configuração podem aparecer nos registros do conector se um plug-in não definir esses valores como secretos. O Kafka Connect trata valores de configuração indefinidos da mesma forma que qualquer outro valor de texto simples.

Se seu plug-in definir uma propriedade como secreta, o Kafka Connect editará o valor da propriedade nos registros do conector. Por exemplo, os registros de conectores a seguir demonstram que o valor será substituído por **`[hidden]`** se um plug-in definir `aws.secret.key` como um tipo `PASSWORD`.

```
   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 segredos apareçam nos arquivos de log do conector, um desenvolvedor de plug-ins deve usar a constante de enumeração `ConfigDef.Type.PASSWORD` do Kafka Connect para definir propriedades confidenciais. Quando uma propriedade for do tipo `ConfigDef.Type.PASSWORD`, o Kafka Connect excluirá seu valor dos registros do conector, mesmo que o valor seja enviado como texto simples.

Monitoramento do Amazon MSK Connect

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho do MSK Connect e de suas outras AWS soluções. A Amazon CloudWatch monitora seus AWS recursos e os aplicativos nos quais você executa AWS em tempo real. Você pode coletar e rastrear métricas, criar painéis personalizados e definir alarmes que o notificam ou que realizam ações quando uma métrica especificada atinge um limite definido. Por exemplo, você pode CloudWatch monitorar o uso da CPU ou outras métricas do seu conector, para que você possa aumentar sua capacidade, se necessário. Para obter mais informações, consulte o Guia CloudWatch do usuário da Amazon.

Você pode usar as seguintes operações de API:

- `DescribeConnectorOperation`: monitore o status das operações de atualização do conector.
- `ListConnectorOperations`: Acompanhe as atualizações anteriores executadas no seu conector.

A tabela a seguir mostra as métricas para as quais o MSK Connect envia CloudWatch sob a `ConnectorName` dimensão. O MSK Connect fornece essas métricas por padrão e sem custo adicional. CloudWatch mantém essas métricas por 15 meses, para que você possa acessar informações históricas e ter uma perspectiva melhor sobre o desempenho de seus conectores. Você também pode definir alarmes que observam determinados limites e enviam notificações ou realizam ações quando esses limites são atingidos. Para obter mais informações, consulte o Guia CloudWatch do usuário da Amazon.

Nome da métrica	Descrição
`CpuUtilization`	O percentual de consumo de CPU por sistema e usuário.
`ErroredTaskCount`	O número de tarefas que apresentaram erro.
`MemoryUtilization`	O percentual da memória total em uma instância de agente, não apenas a memória de pilha da máquina virtual Java (JVM) atualment e em uso. Normalmente, a JVM não libera memória de volta para o sistema operacion

Nome da métrica	Descrição
	al. Portanto, o tamanho da pilha da JVM (MemoryUtilization) geralmente começa com um tamanho mínimo de pilha que aumenta incrementalmente até um máximo estável de cerca de 80-90%. O uso da pilha da JVM pode aumentar ou diminuir conforme o uso efetivo da memória do conector muda.
`RebalanceCompletedTotal`	O número total de rebalanceamentos concluídos por esse conector.
`RebalanceTimeAvg`	O tempo médio em milissegundos gasto pelo conector no rebalanceamento.
`RebalanceTimeMax`	O tempo máximo em milissegundos gasto pelo conector no rebalanceamento.
`RebalanceTimeSinceLast`	O tempo em milissegundos desde que esse conector concluiu o rebalanceamento mais recente.
`RunningTaskCount`	O número de tarefas em execução no conector.
`SinkConsumerByteRate`	O número médio de bytes consumidos por segundo pelo consumidor Sink da estrutura Kafka Connect antes que qualquer transformação seja aplicada aos dados.
`SinkRecordReadRate`	O número médio de registros lidos por segundo do cluster do Apache Kafka ou do Amazon MSK.
`SinkRecordSendRate`	O número médio de registros que são gerados pelas transformações e enviados ao destino por segundo. Esse número não inclui registros filtrados.

Nome da métrica	Descrição
SourceRecordPollRate	O número médio de registros produzidos ou pesquisados por segundo.
SourceProducerByteRate	O número médio de bytes produzidos por segundo pelo produtor de código-fonte da estrutura Kafka Connect após qualquer transformação ser aplicada aos dados.
SourceRecordWriteRate	O número médio de registros gerados pelas transformações e gravados no cluster do Apache Kafka ou do Amazon MSK por segundo.
TaskStartupAttemptsTotal	O número total de inicializações de tarefas que o conector tentou realizar. Você pode usar essa métrica para identificar anomalias nas tentativas de inicialização de tarefas.
TaskStartupSuccessPercentage	O percentual médio de tarefas bem-sucedidas iniciadas para o conector. Você pode usar essa métrica para identificar anomalias nas tentativa s de inicialização de tarefas.
WorkerCount	O número de operadores em execução no conector.
BytesInPerSec	Bytes de metadados transferidos para a estrutura do Kafka Connect para comunicação entre trabalhadores.
BytesOutPerSec	Bytes de metadados transferidos da estrutura do Kafka Connect para comunicação entre trabalhadores.

Exemplos para configurar os recursos do Amazon MSK Connect

Esta seção inclui exemplos para ajudar você a configurar os recursos do Amazon MSK Connect, como conectores e provedores de configuração terceirizados comuns.

Tópicos

Configurar o conector de coletor do Amazon S3

Este exemplo mostra como usar o conector coletor Confluent Amazon S3 e como criar um conector coletor Amazon S3 AWS CLI no MSK Connect.

1. Copie e cole o JSON a seguir em um novo arquivo. Substitua as sequências de caracteres de espaço reservado por valores que correspondam à string de conexão dos servidores bootstrap do seu cluster Amazon MSK e à sub-rede e ao grupo de segurança do cluster. IDs Para obter mais informações sobre como configurar um perfil de execução de serviços, consulte the section called “Perfis e políticas do IAM”.

```
{
    "connectorConfiguration": {
        "connector.class": "io.confluent.connect.s3.S3SinkConnector",
        "s3.region": "us-east-1",
        "format.class": "io.confluent.connect.s3.format.json.JsonFormat",
        "flush.size": "1",
        "schema.compatibility": "NONE",
        "topics": "my-test-topic",
        "tasks.max": "2",
        "partitioner.class":
 "io.confluent.connect.storage.partitioner.DefaultPartitioner",
        "storage.class": "io.confluent.connect.s3.storage.S3Storage",
        "s3.bucket.name": "amzn-s3-demo-bucket"
    },
    "connectorName": "example-S3-sink-connector",
    "kafkaCluster": {
        "apacheKafkaCluster": {
            "bootstrapServers": "<cluster-bootstrap-servers-string>",
```

```
            "vpc": {
                "subnets": [
                    "<cluster-subnet-1>",
                    "<cluster-subnet-2>",
                    "<cluster-subnet-3>"
                ],
                "securityGroups": ["<cluster-security-group-id>"]
            }
        }
    },
    "capacity": {
        "provisionedCapacity": {
            "mcuCount": 2,
            "workerCount": 4
        }
    },
    "kafkaConnectVersion": "2.7.1",
    "serviceExecutionRoleArn": "<arn-of-a-role-that-msk-connect-can-assume>",
    "plugins": [
        {
            "customPlugin": {
                "customPluginArn": "<arn-of-custom-plugin-that-contains-connector-
code>",
                "revision": 1
            }
        }
    ],
    "kafkaClusterEncryptionInTransit": {"encryptionType": "PLAINTEXT"},
    "kafkaClusterClientAuthentication": {"authenticationType": "NONE"}
}
```

2. Execute o AWS CLI comando a seguir na pasta em que você salvou o arquivo JSON na etapa anterior.

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

Veja a seguir um exemplo da saída que você vai obter ao executar o comando com êxito.

```
{
    "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"
```

```
}
```

Configure o conector de coletor EventBridge Kafka para o MSK Connect

Este tópico mostra como configurar o conector do coletor EventBridge Kafka para o MSK Connect. Esse conector permite que você envie eventos do seu cluster MSK para barramentos de EventBridge eventos. Este tópico descreve o processo para criar os recursos necessários e configurar o conector para permitir um fluxo de dados contínuo entre Kafka e. EventBridge

Tópicos

Pré-requisitos

Antes de implantar o conector, verifique se você tem os seguintes recursos:

- Cluster Amazon MSK: um cluster MSK ativo para produzir e consumir mensagens do Kafka.
- Ônibus de EventBridge eventos da Amazon: um ônibus de EventBridge eventos para receber eventos dos tópicos de Kafka.
- Funções do IAM: crie funções do IAM com as permissões necessárias para o MSK Connect e o EventBridge conector.
- Acesso à Internet pública a partir do MSK Connect ou de um endpoint EventBridge de interface VPC criado na VPC e na sub-rede do seu cluster MSK. Isso ajuda você a evitar a passagem pela Internet pública sem a necessidade de gateways NAT.
- Uma máquina cliente, como uma EC2 instância da Amazon ou AWS CloudShell, para criar tópicos e enviar registros para o Kafka.

Configurar os recursos necessários para o MSK Connect

Você cria uma função do IAM para o conector e, em seguida, cria o conector. Você também cria uma EventBridge regra para filtrar os eventos do Kafka enviados para o EventBridge ônibus de eventos.

Tópicos

Função do IAM para o conector

A função do IAM que você associa ao conector deve ter a PutEventspermissão para permitir o envio de eventos para EventBridge. O exemplo de política do IAM a seguir concede a você a permissão para enviar eventos para um barramento de eventos chamado. `example-event-bus` Certifique-se de substituir o ARN do recurso no exemplo a seguir pelo ARN do seu ônibus 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"
    }
  ]
}
```

Além disso, você deve garantir que sua função do IAM para o conector contenha a seguinte política de confiança.

JSON

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

```
        "Service": "kafkaconnect.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Uma EventBridge regra para eventos recebidos

Você cria regras que combinam eventos recebidos com critérios de dados de eventos, conhecidos como padrão de eventos. Com um padrão de evento, você pode definir os critérios para filtrar os eventos recebidos e determinar quais eventos devem acionar uma regra específica e, posteriormente, ser roteados para um destino designado. O exemplo a seguir de um padrão de evento corresponde aos eventos do Kafka enviados para o EventBridge barramento de eventos.

```
{
  "detail": {
    "topic": ["msk-eventbridge-tutorial"]
  }
}
```

Veja a seguir um exemplo de um evento enviado do Kafka para EventBridge usar o conector do coletor 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"
    }
  }
}
```

No EventBridge console, crie uma regra no barramento de eventos usando esse padrão de exemplo e especifique um destino, como um grupo de CloudWatch registros. O EventBridge console configurará automaticamente a política de acesso necessária para o grupo CloudWatch Registros.

Criar o conector

Na seção a seguir, você cria e implanta o conector coletor EventBridge Kafka usando o. AWS Management Console

Tópicos

- Etapa 1: baixar o conector
- Etapa 2: criar um bucket do Amazon S3
- Etapa 3: criar um plug-in no MSK Connect
- Etapa 4: criar o conector

Etapa 1: baixar o conector

Baixe o coletor de EventBridge conectores JAR mais recente na página de GitHub lançamentos do conector EventBridge Kafka. Por exemplo, para baixar a versão v1.4.1, escolha o link do arquivo JAR,`kafka-eventbridge-sink-with-dependencies.jar`, para baixar o conector. Em seguida, salve o arquivo em um local preferido em sua máquina.

Etapa 2: criar um bucket do Amazon S3

1. Para armazenar o arquivo JAR no Amazon S3 para uso com o MSK Connect, abra AWS Management Console o e escolha Amazon S3.
2. No console do Amazon S3, escolha Create bucket e insira um nome de bucket exclusivo. Por exemplo, .**`amzn-s3-demo-bucket1-eb-connector`**

3. Escolha uma região apropriada para seu bucket do Amazon S3. Certifique-se de que corresponda à região em que seu cluster MSK está implantado.
4. Para as configurações do Bucket, mantenha as seleções padrão ou ajuste conforme necessário.
5. Selecione Create bucket (Criar bucket)
6. Faça o upload do arquivo JAR para o bucket do Amazon S3.

Etapa 3: criar um plug-in no MSK Connect

1. Abra o e AWS Management Console, em seguida, navegue até o MSK Connect.
2. No painel de navegação esquerdo, escolha Plugins personalizados.
3. Escolha Criar plug-in e, em seguida, insira o nome do plug-in. Por exemplo, .**`eventbridge-sink-plugin`**
4. Em Localização personalizada do plug-in, cole a URL do objeto S3.
5. Adicione uma descrição opcional para o plug-in.
6. Escolha Criar plug-in.

Depois que o plug-in for criado, você poderá usá-lo para configurar e implantar o conector EventBridge Kafka no MSK Connect.

Etapa 4: criar o conector

Antes de criar o conector, recomendamos criar o tópico necessário do Kafka para evitar erros no conector. Para criar o tópico, use sua máquina cliente.

1. No painel esquerdo do console MSK, escolha Conectores e, em seguida, escolha Criar conector.
2. Na lista de plug-ins, escolha eventbridge-sink-plugin e escolha Próximo.
3. Para o nome do conector, insira**`EventBridgeSink`**.
4. Na lista de clusters, escolha seu cluster MSK.
5. Copie a seguinte configuração para o conector e cole-a no campo Configuração do conector

   Substitua os espaços reservados na configuração a seguir, conforme necessário.

   - Remova `aws.eventbridge.endpoint.uri` se seu cluster MSK tiver acesso público à Internet.

- Se você costuma PrivateLink se conectar com segurança do MSK a EventBridge, substitua a parte DNS depois `https://` pelo nome DNS privado correto do endpoint da interface VPC (opcional) criado anteriormente. EventBridge
- Substitua o ARN do barramento de EventBridge eventos na configuração a seguir pelo ARN do seu barramento de eventos.
- Atualize todos os valores específicos da região.

```
{
  "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 obter mais informações sobre a configuração do conector, consulte eventbridge-kafka-connector.

Se necessário, altere as configurações dos trabalhadores e o escalonamento automático. Também recomendamos usar a versão mais recente disponível (recomendada) do Apache Kafka Connect no menu suspenso. Em Permissões de acesso, use a função criada anteriormente. Também recomendamos ativar o registro em para fins de CloudWatch observabilidade e solução de problemas. Ajuste as outras configurações opcionais, como tags, de acordo com suas necessidades. Em seguida, implante o conector e aguarde até que o status entre no estado Executando.

Envie mensagens para Kafka

Você pode configurar codificações de mensagens, como Apache Avro e JSON, especificando diferentes conversores usando `value.converter` e, opcionalmente, as configurações disponíveis no Kafka Connect. `key.converter`

O connector example neste tópico está configurado para funcionar com mensagens codificadas em JSON, conforme indicado pelo uso de for. `org.apache.kafka.connect.json.JsonConverter` `value converter` Quando o conector estiver no estado Executando, envie registros para o tópico `msk-eventbridge-tutorial` Kafka da sua máquina cliente.

Usar o conector de origem Debezium com provedor de configuração

Este exemplo mostra como usar o plug-in do conector Debezium para MySQL com um banco de dados Amazon Aurora compatível com MySQL como origem. Neste exemplo, também configuramos o AWS Secrets Manager Config Provider de código aberto para externalizar as credenciais do banco de dados no AWS Secrets Manager. Para saber mais sobre os provedores de configuração, consulte Tutorial: Externalizar informações confidenciais usando provedores de configuração.

> **Important**
>
> O plug-in do conector Debezium para MySQL é compatível com apenas uma tarefa e não funciona com o modo de capacidade de escalabilidade automática para o Amazon MSK Connect. Em vez disso, você deve usar o modo de capacidade provisionada e definir `workerCount` igual a um na configuração do conector. Para saber mais sobre os modos de capacidade do MSK Connect, consulte Saiba mais sobre a capacidade de conectores.

Pré-requisitos concluídos para usar o conector de origem Debezium

Seu conector deve ser capaz de acessar a Internet para poder interagir com serviços como os AWS Secrets Manager que estão fora do seu Amazon Virtual Private Cloud. As etapas desta seção ajudam você a concluir as tarefas a seguir para habilitar o acesso à Internet.

- Configure uma sub-rede pública que hospede um gateway NAT e roteie o tráfego para um gateway da Internet em sua VPC.
- Crie uma rota padrão que direcione seu tráfego de sub-rede privada para seu gateway NAT.

Para obter mais informações, consulte Habilitar o acesso à internet para o Amazon MSK Connect.

Pré-requisitos

Antes de habilitar o acesso à Internet, você precisa dos seguintes itens:

- O ID do Amazon Virtual Private Cloud (VPC) associado ao seu cluster. Por exemplo, vpc-123456ab.
- A IDs das sub-redes privadas em sua VPC. Por exemplo, subnet-a1b2c3de, subnet-f4g5h6ij etc. Você deve configurar seu conector com sub-redes privadas.

Para habilitar o acesso à Internet para seu conector

1. Abra o Amazon Virtual Private Cloud console em https://console.aws.amazon.com/vpc/.
2. Crie uma sub-rede pública para seu gateway NAT com um nome descritivo e anote o ID da sub-rede. Para obter instruções detalhadas, consulte Criar uma sub-rede na VPC.
3. Crie um gateway da Internet para que a VPC possa se comunicar com a Internet e anote o ID do gateway. Anexe o gateway da internet à sua VPC. Para obter mais instruções, consulte Criar e anexar um gateway da Internet à VPC.
4. Provisione um gateway NAT público para que os hosts em suas sub-redes privadas possam acessar sua sub-rede pública. Ao criar o gateway NAT, selecione a sub-rede pública que você criou anteriormente. Para obter instruções, consulte Create a NAT gateway (Criar um gateway NAT)
5. Configure suas tabelas de rotas. Para concluir essa configuração, você deve ter duas tabelas de rotas no total. Você já deve ter uma tabela de rotas principal criada automaticamente junto com sua VPC. Nesta etapa, você cria uma tabela de rotas adicional para sua sub-rede pública.

    a. Use as configurações a seguir para modificar a tabela de rotas principal da sua VPC para que suas sub-redes privadas roteiem o tráfego para seu gateway NAT. Para obter instruções, consulte Trabalhar com tabelas de rotas no Guia do usuário do Amazon Virtual Private Cloud.

    Tabela de rotas MSKC privado

Propriedade	Valor
Name tag	Recomendamos que você atribua uma tag de nome descritivo a essa tabela de

Propriedade	Valor
	rotas para ajudar na identificação dela. Por exemplo, MSKC privado.
Sub-redes associadas	Suas sub-redes privadas
Uma rota para habilitar o acesso à Internet para o MSK Connect	• Destino: 0.0.0.0/0 • Alvo: o ID do seu gateway NAT Por exemplo, nat-12a345bc6789efg1h.
Uma rota local para o tráfego interno	• Destino: 10.0.0.0/16 Esse valor pode ser diferente dependendo do bloco CIDR da sua VPC. • Alvo: local

b. Siga as instruções em Criar uma tabela de rotas personalizada para criar uma tabela de rotas para sua sub-rede pública. Ao criar a tabela, insira um nome descritivo no campo Tag de nome para ajudar você a identificar a qual sub-rede a tabela está associada. Por exemplo, MSKC público.

c. Configure sua tabela de rotas MSKC público usando as configurações a seguir.

Propriedade	Valor
Name tag	MSKC público ou um nome descritivo diferente que você escolher
Sub-redes associadas	Sua sub-rede pública com gateway NAT
Uma rota para habilitar o acesso à Internet para o MSK Connect	• Destino: 0.0.0.0/0 • Alvo: o ID do seu gateway da Internet Por exemplo, igw-1a234bc5.
Uma rota local para o tráfego interno	• Destino: 10.0.0.0/16 Esse valor pode ser diferente dependendo do bloco CIDR da sua VPC. • Alvo: local

Agora que habilitou o acesso à Internet para o Amazon MSK Connect, você está pronto para criar um conector.

Criar um conector de origem Debezium

Este procedimento descreve como criar um conector de origem Debezium.

1. Criar um plug-in personalizado

   a. Baixe o plug-in do conector MySQL para obter a versão estável mais recente no site do Debezium. Anote a versão do Debezium que você baixou (versão 2.x ou a antiga série 1.x). Você criará um conector com base na sua versão do Debezium mais adiante neste procedimento.

   b. Baixe e extraia o AWS Secrets Manager Config Provider.

   c. Coloque os seguintes arquivos no mesmo diretório:

      - A pasta `debezium-connector-mysql`.
      - A pasta `jcusten-border-kafka-config-provider-aws-0.1.1`.

   d. Compacte em um arquivo ZIP o diretório que você criou na etapa anterior e, em seguida, carregue o arquivo ZIP em um bucket do S3. Para obter instruções, consulte Upload de objetos no Guia do usuário do Amazon S3.

   e. Copie e cole o JSON a seguir em um arquivo. Por exemplo, `.debezium-source-custom-plugin.json` *<example-custom-plugin-name>*Substitua pelo nome que você deseja que o plug-in tenha, *<amzn-s3-demo-bucket-arn>* pelo ARN do bucket do Amazon S3 em que você fez o upload do arquivo ZIP *<file-key-of-ZIP-object>* e pela chave de arquivo do objeto ZIP que você carregou no S3.

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

f. Execute o AWS CLI comando a seguir na pasta em que você salvou o arquivo JSON para criar um plug-in.

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

Você deve ver uma saída semelhante ao seguinte exemplo.

```
{
    "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. Execute o comando a seguir para verificar o estado do plug-in. O estado do cluster deve mudar de `CREATING` para `ACTIVE`. Substitua o espaço reservado de ARN pelo ARN que você obteve na saída do comando anterior.

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

2. Configure AWS Secrets Manager e crie um segredo para suas credenciais de banco de dados

a. Abra o console do Secrets Manager em https://console.aws.amazon.com/secretsmanager/.

b. Crie um novo segredo para armazenar as credenciais de login do banco de dados. Para obter instruções, consulte Criar um segredo no Guia do usuário do AWS Secrets Manager.

c. Copie o ARN do seu segredo.

d. Adicione as permissões do Secrets Manager do exemplo de política a seguir ao seu Saiba mais sobre o perfil de execução do serviço. *<arn:aws:secretsmanager:us-east-1:123456789000:secret:MySecret-1234>*Substitua pelo ARN do seu segredo.

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 obter instruções sobre como adicionar permissões do IAM, consulte Adicionar e remover permissões de identidade do IAM no Guia do usuário do IAM.

3. Criar uma configuração personalizada de operador com informações sobre seu provedor de configuração

   a. Copie as seguintes propriedades de configuração do operador em um arquivo, substituindo as strings de espaço reservado por valores que correspondam ao seu cenário. Para saber mais sobre as propriedades de configuração do AWS Secrets Manager Config Provider, consulte a SecretsManagerConfigProviderdocumentação do plug-in.

   ```
   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. Execute o AWS CLI comando a seguir para criar sua configuração de trabalhador personalizada.

   Substitua os valores a seguir:

   - *<my-worker-config-name>*- um nome descritivo para sua configuração de trabalhador personalizada
   - *<encoded-properties-file-content-string>*- uma versão codificada em base64 das propriedades de texto simples que você copiou na etapa anterior

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

4. Criar um conector

   a. Copie o seguinte JSON que corresponde à sua versão do Debezium (2.x ou 1.x) e cole-o em um novo arquivo. Substitua as strings *<placeholder>* por valores que correspondam ao seu cenário. Para obter mais informações sobre como configurar um perfil de execução de serviços, consulte the section called “Perfis e políticas do IAM”.

   Para especificar as credenciais do banco de dados, a configuração usa variáveis como `${secretManager:MySecret-1234:dbusername}` em vez de texto simples. Substitua *MySecret-1234* pelo nome do seu segredo e inclua o nome da chave que você deseja recuperar. Você também deve substituir *<arn-of-config-provider-worker-configuration>* pelo ARN da sua configuração personalizada de operador.

   Debezium 2.x

   Para as versões 2.x do Debezium, copie o seguinte JSON e cole-o em um novo arquivo. Substitua as strings *<placeholder>* por valores que correspondam ao seu cenário.

```
{
 "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 as versões 1.x do Debezium, copie o seguinte JSON e cole-o em um novo arquivo. Substitua as strings *<placeholder>* por valores que correspondam ao seu cenário.

```
{
 "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. Execute o AWS CLI comando a seguir na pasta em que você salvou o arquivo JSON na etapa anterior.

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

Veja a seguir um exemplo da saída que você vai obter ao executar o comando com êxito.

```
{
    "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"
}
```

Atualizar uma configuração do conector Debezium

Para atualizar a configuração do conector Debezium, siga estas etapas:

1. Copie o seguinte JSON e cole-o em um novo arquivo. Substitua as strings <placeholder> por valores que correspondam ao seu cenário.

```
{
   "connectorArn": <connector_arn>,
   "connectorConfiguration": <new_configuration_in_json>,
   "currentVersion": <current_version>
}
```

2. Execute o AWS CLI comando a seguir na pasta em que você salvou o arquivo JSON na etapa anterior.

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

Veja a seguir um exemplo da saída quando você executa o comando com êxito.

```
{
    "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. Agora você pode executar o seguinte comando para monitorar o estado atual da operação:

```
aws kafkaconnect describe-connector-operation --connector-operation-arn
 <operation_arn>
```

Para ver um exemplo de conector Debezium com etapas detalhadas, consulte Introdução ao Amazon MSK Connect: transmita dados de e para seus clusters do Apache Kafka usando conectores gerenciados.

Migrar para o Amazon MSK Connect

Esta seção descreve como migrar a aplicação de conector Apache Kafka para o Amazon Managed Streaming para Apache Kafka Connect (Amazon MSK Connect). Para saber mais sobre os benefícios de migrar para o Amazon MSK Connect, consulte ???.

Esta seção também descreve os tópicos de gerenciamento de estado usados pelo Kafka Connect e pelo Amazon MSK Connect e aborda os procedimentos para migrar conectores de origem e de coletor.

Saiba mais sobre os tópicos internos usados pelo Kafka Connect

Uma aplicação Apache Kafka Connect que está sendo executada no modo distribuído armazena seu estado usando tópicos internos no cluster do Kafka e na associação ao grupo. A seguir estão os valores de configuração que correspondem aos tópicos internos usados nas aplicações do Kafka Connect:

- Tópico de configuração, especificado por meio de `config.storage.topic`

  No tópico de configuração, o Kafka Connect armazena a configuração de todos os conectores e tarefas que foram iniciados pelos usuários. Sempre que os usuários atualizam a configuração de um conector ou quando um conector solicita uma reconfiguração (por exemplo, o conector detecta que pode iniciar mais tarefas), um registro é emitido para esse tópico. Esse tópico tem compactação habilitada, portanto, ele sempre mantém o último estado de cada entidade.
- Tópico de deslocamentos, especificado por meio de `offset.storage.topic`

  No tópico de deslocamentos, o Kafka Connect armazena os deslocamentos dos conectores de origem. Assim como o tópico de configuração, o tópico de deslocamentos está habilitado para

compactação. Esse tópico é usado para gravar as posições de origem somente para conectores de origem que produzem dados para o Kafka de sistemas externos. Os conectores de coletor, que leem dados do Kafka e os enviam para sistemas externos, armazenam os deslocamentos de consumo usando grupos regulares de consumidores do Kafka.

- Tópico de status, especificado por meio de `status.storage.topic`

  No tópico de status, o Kafka Connect armazena o estado atual dos conectores e das tarefas. Esse tópico é usado como o local central para os dados que são consultados pelos usuários da API REST. Esse tópico permite que os usuários consultem qualquer operador e ainda obtenham o status de todos os plug-ins em execução. Assim como os tópicos de configuração e deslocamentos, o tópico de status também está habilitado para compactação.

Além desses tópicos, o Kafka Connect faz uso extensivo da API de associação a grupos do Kafka. Os grupos são recebem o nome de acordo com o nome do conector. Por exemplo, para um conector chamado file-sink, o grupo é nomeado. connect-file-sink Cada consumidor do grupo fornece registros para uma única tarefa. Esses grupos e seus deslocamentos podem ser recuperados usando ferramentas regulares de grupos de consumidores, como `Kafka-consumer-group.sh`. Para cada conector de coletor, o runtime do Connect executa um grupo regular de consumidores que extrai registros do Kafka.

Gerenciamento de estados das aplicações do Amazon MSK Connect

Por padrão, o Amazon MSK Connect cria três tópicos separados no cluster do Kafka para cada conector do Amazon MSK para armazenar a configuração, o deslocamento e o status do conector. Os nomes de tópicos padrão são estruturados da seguinte maneira:

- *`connector-name`*__msk_connect_configs_ _ *`connector-id`*
- *`connector-name`*__msk_connect_status_ _ *`connector-id`*
- *`connector-name`*__msk_connect_offsets_ _ *`connector-id`*

> **Note**
>
> Para fornecer continuidade de deslocamento entre conectores de origem, você pode usar um tópico de deslocamento de armazenamento de sua escolha em vez do tópico padrão. Especificar um tópico de deslocamento de armazenamento ajuda você a realizar tarefas como criar um conector de origem que retoma a leitura desde o último deslocamento de

um conector anterior. Para especificar um tópico de deslocamento de armazenamento, forneça um valor para a propriedade offset.storage.topic em sua configuração de operador do Amazon MSK Connect antes de criar um conector.

Migre conectores de origem para o Amazon MSK Connect

Os conectores de origem são aplicações do Apache Kafka Connect que importam registros de sistemas externos para o Kafka. Esta seção descreve o processo de migração de aplicações de conectores de origem do Apache Kafka Connect que estão sendo executados on-premises ou clusters autogerenciados do Kafka Connect que estão sendo executados na para o Amazon MSK Connect. AWS

A aplicação do conector de origem do Kafka Connect armazena deslocamentos em um tópico nomeado com o valor definido para a propriedade de configuração `offset.storage.topic`. A seguir estão exemplos de mensagens de deslocamento para um conector JDBC que está executando duas tarefas que importam dados de duas tabelas diferentes denominadas `movies` e `shows`. A linha mais recente importada da tabela de filmes tem um ID primário de `18343`. A linha mais recente importada da tabela de shows tem um ID primário de `732`.

```
["jdbcsource",{"protocol":"1","table":"sample.movies"}] {"incrementing":18343}
["jdbcsource",{"protocol":"1","table":"sample.shows"}] {"incrementing":732}
```

Para migrar conectores de origem para o Amazon MSK Connect, faça o seguinte:

1. Crie um plug-in personalizado do Amazon MSK Connect extraindo bibliotecas de conectores do seu cluster do Kafka Connect on-premises ou autogerenciado.
2. Crie propriedades de operador do Amazon MSK Connect e defina as propriedades `key.converter`, `value.converter` e `offset.storage.topic` com os mesmos valores estipulados para o conector do Kafka que está sendo executado em seu cluster atual do Kafka Connect.
3. Pause a aplicação do conector no cluster existente fazendo uma solicitação `PUT /connectors/`*`connector-name`*`/pause` no cluster existente do Kafka Connect.
4. Certifique-se de que todas as tarefas da aplicação do conector estejam completamente interrompidas. Você pode interromper as tarefas fazendo uma solicitação `GET /connectors/`*`connector-name`*`/status` no cluster existente do Kafka Connect ou consumindo as mensagens do nome do tópico definido para a propriedade `status.storage.topic`.

5. Obtenha a configuração do conector do cluster existente. Você pode obter a configuração do conector fazendo uma solicitação `GET /connectors/connector-name/config/` no cluster existente ou consumindo as mensagens do nome do tópico definido para a propriedade `config.storage.topic`.
6. Crie um Amazon MSK Connector com o mesmo nome de um cluster existente. Crie esse conector usando o plug-in personalizado do conector que você criou na etapa 1, as propriedades do operador que você criou na etapa 2 e a configuração do conector que você extraiu na etapa 5.
7. Quando o status do Amazon MSK Connector estiver `active`, visualize os logs para verificar se o conector começou a importar dados do sistema de origem.
8. Exclua o conector no cluster existente fazendo uma solicitação `DELETE /connectors/connector-name`.

Migrar conectores de coletor para o Amazon MSK Connect

Os conectores de coletor são aplicações do Apache Kafka Connect que importam registros de sistemas externos para o Kafka. Esta seção descreve o processo de migração de aplicações de conectores de coletor do Apache Kafka Connect que estão sendo executados on-premises ou clusters autogerenciados do Kafka Connect que estão sendo executados na para o Amazon MSK Connect. AWS

Os conectores de coletor do Kafka Connect usam a API de associação de grupos do Kafka e armazenam deslocamentos nos mesmos tópicos `__consumer_offset` de uma aplicação de consumo típico. Esse comportamento simplifica a migração do conector de coletor de um cluster autogerenciado para o Amazon MSK Connect.

Para migrar conectores de coletor para o Amazon MSK Connect, faça o seguinte:

1. Crie um plug-in personalizado do Amazon MSK Connect extraindo bibliotecas de conectores do seu cluster do Kafka Connect on-premises ou autogerenciado.
2. Crie propriedades de operador do Amazon MSK Connect e defina as propriedades `key.converter` e `value.converter` com os mesmos valores definidos para o conector do Kafka que está sendo executado no cluster existente do Kafka Connect.
3. Pause a aplicação do conector no cluster existente fazendo uma solicitação `PUT /connectors/connector-name/pause` no cluster existente do Kafka Connect.
4. Certifique-se de que todas as tarefas da aplicação do conector estejam completamente interrompidas. Você pode interromper as tarefas fazendo uma solicitação `GET /`

`connectors/connector-name/status` no cluster existente do Kafka Connect ou consumindo as mensagens do nome do tópico definido para a propriedade `status.storage.topic`.

5. Obtenha a configuração do conector do cluster existente. Você pode obter a configuração do conector fazendo uma solicitação `GET /connectors/connector-name/config` no cluster existente ou consumindo as mensagens do nome do tópico definido para a propriedade `config.storage.topic`.
6. Crie um Amazon MSK Connector com o mesmo nome do cluster existente. Crie esse conector usando o plug-in personalizado do conector que você criou na etapa 1, as propriedades do operador que você criou na etapa 2 e a configuração do conector que você extraiu na etapa 5.
7. Quando o status do Amazon MSK Connector estiver `active`, visualize os logs para verificar se o conector começou a importar dados do sistema de origem.
8. Exclua o conector no cluster existente fazendo uma solicitação `DELETE /connectors/connector-name`.

Solução de problemas no Amazon MSK Connect

As informações a seguir podem ajudar você a solucionar problemas que você pode vir a enfrentar com MSK Connect. Você também pode publicar seu problema no AWS re:Post.

O conector não consegue acessar recursos hospedados na Internet pública

Consulte Como habilitar o acesso à Internet para o Amazon MSK Connect.

O número de tarefas em execução do conector não é igual ao número de tarefas especificado em tasks.max

Aqui estão alguns motivos pelos quais um conector pode usar menos tarefas do que o valor especificado na configuração tasks.max:

- Algumas implementações de conectores limitam o número de tarefas que podem ser usadas. Por exemplo, o conector Debezium para MySQL está limitado ao uso de uma única tarefa.
- Ao usar o modo de capacidade com escalabilidade automática, o Amazon MSK Connect substitui a propriedade tasks.max de um conector por um valor proporcional ao número de trabalhadores em execução no conector e ao número de por trabalhador. MCUs
- Para conectores de coletor, o nível de paralelismo (número de tarefas) não pode ser maior que o número de partições de tópicos. Embora você possa definir tasks.max com um valor maior que esse, uma única partição nunca é processada por mais de uma única tarefa por vez.

- No Kafka Connect 2.7.x, o atribuidor de partição de consumidor padrão é `RangeAssignor`. O comportamento desse atribuidor é fornecer a primeira partição de cada tópico a um único consumidor, a segunda partição de cada tópico a um único consumidor etc. Isso significa que o número máximo de tarefas ativas usadas por um conector de coletor com `RangeAssignor` é igual ao número máximo de partições em qualquer tópico que esteja sendo consumido. Se isso não funcionar para seu caso de uso, você deve criar uma configuração de agente na qual a propriedade `consumer.partition.assignment.strategy` seja definida como um atribuidor de partição de consumidor mais adequado. Consulte Interface do Kafka 2.7 ConsumerPartitionAssignor: todas as classes de implementação conhecidas.

O que é o replicador do Amazon MSK?

O Amazon MSK Replicator é um recurso do Amazon MSK que permite replicar dados de forma confiável entre clusters do Amazon MSK em clusters diferentes ou iguais. Região da AWS No entanto, os clusters de origem e de destino devem estar no mesmo Conta da AWS. Com o replicador do MSK, você pode criar facilmente aplicações de streaming regionalmente resilientes para aumentar a disponibilidade e a continuidade dos negócios. O replicador do MSK fornece replicação assíncrona automática em clusters do MSK, eliminando a necessidade de criar código personalizado, gerenciar a infraestrutura ou configurar redes entre regiões.

O replicador do MSK escala automaticamente os recursos subjacentes, permitindo que você replique dados sob demanda sem precisar monitorar ou escalar a capacidade. O MSK Replicator também replica os metadados necessários do Kafka, incluindo configurações de tópicos, listas de controle de acesso () ACLs e compensações de grupos de consumidores. Se ocorrer um evento inesperado em uma região, você pode fazer o failover para a outra AWS região e retomar o processamento sem problemas.

O replicador do MSK é compatível com Cross-Region Replication (CRR – Replicação entre regiões) e Same-Region Replication (SRR – Replicação na mesma região). Na replicação entre regiões, os clusters MSK de origem e de destino estão em regiões diferentes. AWS Na replicação na mesma região, os clusters MSK de origem e de destino estão na mesma região. AWS Você precisa criar clusters de origem e de destino do MSK antes de usá-los com o replicador do MSK.

> **Note**
>
> O MSK Replicator suporta as seguintes AWS regiões: Leste dos EUA (us-east-1, Norte da Virgínia); Leste dos EUA (us-east-2, Ohio); Oeste dos EUA (us-west-2, Oregon); Europa (eu-west-1, Irlanda); Europa (eu-central-1, Frankfurt); Ásia-Pacífico (ap-southeast-1 Ásia-Pacífico, Cingapura); Ásia-Pacífico (ap-southeast-2, Sydney), Europa (eu-north-1, Estocolmo), Ásia-Pacífico (ap-south-1, Mumbai), Europa (eu-west-3, Paris), América do Sul (sa-east-1, São Paulo), Ásia Pacífico (ap-northeast-2, Seul), Europa (eu-west-2, Londres), Ásia-Pacífico (ap-northeast-1, Tóquio), Oeste dos EUA (us-west-1, Norte da Califórnia), Canadá (ca-central-1, Central).

Veja alguns usos comuns do Replicador do Amazon MSK.

- Crie aplicações de streaming multirregionais: crie aplicações de streaming altamente disponíveis e tolerantes a falhas para aumentar a resiliência sem configurar soluções personalizadas.
- Acesso a dados com menor latência: forneça acesso a dados com menor latência para consumidores em diferentes regiões geográficas.
- Distribua dados para seus parceiros: copie dados de um cluster do Apache Kafka para vários clusters do Apache Kafka, para que os diferentes teams/partners tenham suas próprias cópias dos dados.
- Agregar dados para analytics: copie dados de vários clusters do Apache Kafka em um cluster para gerar facilmente insights sobre dados agregados em tempo real.
- Escreva localmente, acesse seus dados globalmente: configure a replicação multiativa para propagar automaticamente as gravações realizadas em uma AWS região para outras regiões, fornecendo dados com menor latência e custo.

Funcionamento do replicador do Amazon MSK

Para começar a usar o MSK Replicator, você precisa criar um novo replicador na região do seu cluster de destino. AWS O MSK Replicator copia automaticamente todos os dados do cluster na AWS região primária chamada origem para o cluster na região de destino chamada destino. Os clusters de origem e de destino podem estar na mesma região ou em AWS regiões diferentes. Você precisará criar o cluster de destino se ele não existir.

Quando você cria um replicador, o MSK Replicator implanta todos os recursos necessários na AWS região do cluster de destino para otimizar a latência da replicação de dados. A latência de replicação varia com base em muitos fatores, incluindo a distância da rede entre as AWS regiões dos seus clusters MSK, a capacidade de taxa de transferência dos clusters de origem e de destino e o número de partições nos clusters de origem e de destino. O replicador do MSK escala automaticamente os recursos subjacentes, permitindo que você replique dados sob demanda sem precisar monitorar ou escalar a capacidade.

Replicação de dados

Por padrão, o Replicador do MSK copia todos os dados de maneira assíncrona do deslocamento mais recente nas partições de tópicos do cluster de origem para o cluster de destino. Se a configuração “Detectar e copiar novos tópicos” estiver ativada, o Replicador do MSK detectará e copiará automaticamente novos tópicos ou partições de tópicos para o cluster de destino. No entanto, pode levar até 30 segundos para que o replicador detecte e crie os novos tópicos ou

partições de tópicos no cluster de destino. Qualquer mensagem produzida no tópico de origem antes da criação do tópico no cluster de destino não será replicada. Como alternativa, você pode configurar o replicador durante a criação para iniciar a replicação a partir do primeiro deslocamento nas partições de tópicos do cluster de origem, caso queira replicar as mensagens existentes nos tópicos para o cluster de destino.

O Replicador do MSK não armazena seus dados. Os dados são consumidos do cluster de origem, armazenados em buffer na memória e gravados no cluster de destino. O buffer é limpo automaticamente quando os dados são gravados com êxito ou falham após novas tentativas. Toda a comunicação e os dados entre o Replicador do MSK e os clusters são sempre criptografados em trânsito. Todas as chamadas da API do MSK Replicator`DescribeClusterV2`, como,`CreateTopic`, `DescribeTopicDynamicConfiguration` são capturadas em. AWS CloudTrail Os logs do agente do MSK também refletirão o mesmo.

O Replicador do MSK cria tópicos no cluster de destino com um fator de replicação de 3. Se necessário, você pode modificar o fator de replicação diretamente no cluster de destino.

Replicação de metadados

O Replicador do MSK também é compatível com a cópia dos metadados do cluster de origem para o cluster de destino. Os metadados incluem configuração de tópicos, listas de controle de acesso (ACLs) e compensações de grupos de consumidores. Assim como a replicação de dados, a replicação de metadados também ocorre de forma assíncrona. Para uma melhor performance, o Replicador do MSK prioriza a replicação de dados sobre a replicação de metadados.

A tabela a seguir é uma lista das listas de controle de acesso (ACLs) que o MSK Replicator copia.

Operação	Pesquisa	APIs permitido
Alter	Tópico	CreatePartitions
AlterConfigs	Tópico	AlterConfigs
Criar	Tópico	CreateTopics, Metadados
Excluir	Tópico	DeleteRecords, DeleteTopics

Operação	Pesquisa	APIs permitido
Descrever	Tópico	ListOffsets, Metadados ,, OffsetFetch OffsetFor LeaderEpoch
DescribeConfigs	Tópico	DescribeConfigs
Leitura	Tópico	Busque,, OffsetCommit TxnOffsetCommit
Write (deny only)	Tópico	Produzir, AddPartitionsToTxn

O MSK Replicator copia o tipo de padrão LITERAL ACLs somente para o tipo de recurso Topic. O tipo de padrão PREFIXADO ACLs e outro tipo de recurso não ACLs são copiados. O MSK Replicator também não exclui ACLs no cluster de destino. Se você excluir uma ACL no cluster de origem, também deverá excluir no cluster de destino ao mesmo tempo. Para obter mais detalhes sobre os ACLs recursos, padrões e operações do Kafka, consulte https://kafka.apache.org/documentation/#security_authz_cli.

O MSK Replicator replica somente o Kafka ACLs, que o controle de acesso do IAM não usa. Se seus clientes estão usando o controle de acesso do IAM read/write aos seus clusters do MSK, você também precisa configurar as políticas relevantes do IAM no cluster de destino para um failover contínuo. Isso também é válido para configurações de replicação de nomes de tópicos prefixados e idênticos.

Como parte da sincronização de deslocamentos de grupos de consumidores, o Replicador do MSK otimiza para os consumidores no cluster de origem, que estão lendo de uma posição mais próxima à ponta do stream (final da partição do tópico). Se os grupos de consumidores estiverem em atraso no cluster de origem, você poderá observar um atraso maior para esses grupos de consumidores no destino em comparação com a origem. Isso significa que, após o failover para o cluster de destino, os consumidores reprocessarão mais mensagens duplicadas. Para reduzir esse atraso, os consumidores no cluster de origem precisariam se atualizar e começar a consumir a partir da ponta do stream (final da partição do tópico). À medida que os consumidores se atualizarem, o Replicador do MSK reduzirá automaticamente o atraso.

Configuração do nome do tópico

O Replicador do MSK tem dois modos de configuração de nomes de tópicos: Prefixado (padrão) ou replicação de nomes de tópicos Idênticos.

Replicação de nomes de tópicos prefixados

Por padrão, o Replicador do MSK cria tópicos no cluster de destino com um prefixo gerado automaticamente adicionado ao nome do tópico do cluster de destino, como `<sourceKafkaClusterAlias>.topic`. Isso serve para distinguir os tópicos replicados de outros no cluster de destino e para evitar a replicação circular de dados entre os clusters.

Por exemplo, o MSK Replicator replica dados em um tópico chamado “tópico” do cluster de origem para um novo tópico no cluster de destino chamado **<** Alias**>**.topic. sourceKafkaCluster Você pode encontrar o prefixo que será adicionado aos nomes dos tópicos no cluster de destino no campo sourceKafkaClusterAlias usando a `DescribeReplicator` API ou a página de detalhes do Replicator no console do MSK. O prefixo no cluster de destino é **<** sourceKafkaCluster Alias**>**.

Para garantir que os consumidores possam reiniciar o processamento de maneira confiável diretamente do cluster em espera, você precisa configurar os consumidores para ler os dados dos tópicos usando um operador curinga `.*`. Por exemplo, seus consumidores precisariam consumir usando. `*topic1`em ambas as AWS regiões. Esse exemplo também pode incluir um tópico como `footopic1`, portanto, ajuste o operador curinga de acordo com suas necessidades.

Você deve usar o Replicador do MSK, que adicionará um prefixo quando você quiser manter os dados do replicador em um tópico separado no cluster de destino, como para configurações de cluster ativo-ativo.

Replicação de nomes de tópicos idênticos

Como alternativa à configuração padrão, o Replicador do Amazon MSK permite que você crie um replicador com a replicação de tópicos definida como replicação de nomes de tópicos idênticos (mantenha o mesmo nome de tópicos no console). Você pode criar um novo replicador na AWS região que tenha seu cluster MSK de destino. Tópicos replicados com nomes idênticos permitem que você evite reconfigurar clientes para ler tópicos replicados.

A replicação de nomes de tópicos idênticos (mantenha o mesmo nome de tópicos no console) tem as seguintes vantagens:

- Permite que você mantenha nomes de tópicos idênticos durante o processo de replicação, além de evitar automaticamente o risco de loops de replicação infinitos.

- Simplifica a configuração e a operação de arquiteturas de streaming de vários clusters, pois você pode evitar a reconfiguração de clientes para ler os tópicos replicados.
- Para arquiteturas de cluster ativo-passivo, a funcionalidade de replicação de nomes de tópicos idênticos também simplifica o processo de failover, permitindo que as aplicações façam o failover facilmente para um cluster em espera sem exigir nenhuma alteração no nome do tópico ou reconfiguração do cliente.
- Pode ser usado para consolidar com mais facilidade dados de vários clusters do MSK em um único cluster para agregação de dados ou analytics centralizado. Isso exige que você crie replicadores separados para cada cluster de origem e para o mesmo cluster de destino.
- Pode simplificar a migração de dados de um cluster do MSK para outro replicando dados para tópicos de nomes idênticos no cluster de destino.

O Replicador do Amazon MSK usa cabeçalhos do Kafka para evitar automaticamente que os dados sejam replicados de volta ao tópico de origem, eliminando o risco de ciclos infinitos durante a replicação. Um cabeçalho é um par de chave-valor que pode ser incluído com a chave, o valor e o carimbo de data e hora em cada mensagem do Kafka. O Replicador do MSK incorpora identificadores para o cluster e o tópico de origem no cabeçalho de cada registro que está sendo replicado. O Replicador do MSK usa as informações do cabeçalho para evitar loops de replicação infinitos. Você deve verificar se os clientes conseguem ler os dados replicados conforme o esperado.

Tutorial: Configurar clusters de origem e destino para o Replicador do Amazon MSK

Este tutorial mostra como configurar um cluster de origem e um cluster de destino na mesma AWS região ou em AWS regiões diferentes. Posteriormente, você também pode usar esses clusters para criar um replicador do Amazon MSK.

Preparar o cluster de origem do Amazon MSK

Se você já tiver um cluster de origem do MSK criado para o replicador do MSK, certifique-se de que ele atenda aos requisitos descritos nesta seção. Caso contrário, siga estas etapas para criar um cluster de origem com a tecnologia sem servidor ou provisionado do MSK.

O processo de criação de um cluster de origem do replicador do MSK entre regiões e na mesma região é semelhante. As diferenças estão nas chamadas nos procedimentos a seguir.

1. Crie um cluster provisionado ou com tecnologia sem servidor do MSK com o controle de acesso do IAM ativado na região de origem. Seu cluster de origem deve ter, no mínimo, três agentes.
2. Para um replicador do MSK entre regiões, se a origem for um cluster provisionado, configure-o com a conectividade privada multi-VPC ativada para esquemas de controle de acesso do IAM. Observe que não há compatibilidade com o tipo de autenticação não autenticado quando o recurso multi-VPC estiver ativado. Você não precisa ativar a conectividade privada de várias VPCs para outros esquemas de autenticação (mTLS) ou esquemas de SASL/SCRAM). You can simultaneously use mTLS or SASL/SCRAM autenticação para seus outros clientes que se conectam ao seu cluster MSK. Você pode configurar a conectividade privada multi-VPC nos detalhes do cluster no console, nas Configurações de rede ou com a API `UpdateConnectivity`. Consulte Proprietário do cluster ativa o recurso multi-VPC. Se seu cluster de origem for um cluster do MSK Serverless, você não precisará ativar a conectividade privada multi-VPC.

   Para um replicador do MSK na mesma região, o cluster de origem do MSK não exige conectividade privada multi-VPC e o cluster ainda pode ser acessado por outros clientes usando o tipo de autenticação não autenticada.
3. Para replicadores do MSK entre regiões, você deve anexar uma política de permissões baseada em recursos ao cluster de origem. Isso permite que o MSK se conecte a esse cluster para replicar dados. Você pode fazer isso usando os procedimentos da CLI ou AWS do console abaixo. Veja também as Políticas baseadas em recursos do Amazon MSK. Não há necessidade de executar essa etapa para replicadores do MSK na mesma região.

Console: create resource policy

Atualize a política de cluster de origem com o seguinte JSON. Substitua o espaço reservado pelo ARN do cluster de origem.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Service": [
                "kafka.amazonaws.com"
            ]
```

```
        },
        "Action": [
            "kafka:CreateVpcConnection",
            "kafka:GetBootstrapBrokers",
            "kafka:DescribeClusterV2"
        ],
        "Resource": "arn:aws:kafka:us-
east-1:123456789012:cluster/myCluster/abcd1234-5678-90ab-cdef-1234567890ab-1"
    }
  ]
}
```

Use a opção Editar política de cluster no menu Ações na página de detalhes do cluster.

CLI: create resource policy

Observação: se você usar o AWS console para criar um cluster de origem e escolher a opção de criar uma nova função do IAM, AWS anexe a política de confiança necessária à função. Se você quiser que o MSK use um perfil existente do IAM ou se você criar um perfil, anexe as seguintes políticas de confiança a esse perfil para que o replicador do MSK possa assumi-lo. Para obter informações sobre como modificar a relação de confiança de um perfil, consulte Modificação de um perfil.

1. Obtenha a versão atual da política de cluster do MSK usando esse comando. Substitua os espaços reservados pelo ARN efetivo do cluster.

```
aws kafka get-cluster-policy –cluster-arn <Cluster ARN>
{
"CurrentVersion": "K1PA6795UKM GR7",
"Policy": "..."
}
```

2. Crie uma política baseada em recursos para permitir que o replicador do MSK acesse seu cluster de origem. Use a sintaxe a seguir como modelo, substituindo o espaço reservado pelo ARN efetivo do cluster de origem.

```
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>"
}
]
```

Preparar o cluster de destino do Amazon MSK

Crie um cluster de destino do MSK (provisionado ou com tecnologia sem servidor) com o controle de acesso do IAM ativado. O cluster de destino não exige que a conectividade privada multi-VPC esteja ativada. O cluster de destino pode estar na mesma AWS região ou em uma região diferente do cluster de origem. Os clusters de origem e de destino devem estar na mesma AWS conta. Seu cluster de destino deve ter, no mínimo, três agentes.

Tutorial: Criar um Replicador do Amazon MSK

Depois de configurar os clusters de origem e de destino, você pode usá-los para criar um Replicador do Amazon MSK. Antes de criar o replicador do Amazon MSK, certifique-se de ter Permissões necessárias do IAM para criar um Replicador do MSK.

Tópicos

- Escolher seu cluster de destino
- Definir configurações e permissões do replicador

Considerações sobre a criação de um Replicador do Amazon MSK

As seções a seguir fornecem uma visão geral dos pré-requisitos, das configurações compatíveis e das práticas recomendadas para o uso do recurso Replicador do MSK. Ele abrange as permissões necessárias, a compatibilidade do cluster e os requisitos específicos da tecnologia sem servidor, bem como orientações sobre o gerenciamento do replicador após a criação.

Permissões necessárias do IAM para criar um Replicador do MSK

Veja um exemplo da política do IAM necessária para criar um replicador do MSK. A ação `kafka:TagResource` só é necessária se as tags forem fornecidas ao criar o replicador do MSK. As políticas do replicador do IAM devem ser anexadas ao perfil do IAM correspondente ao seu cliente. Para obter informações sobre a criação de políticas de autorização, consulte Criar políticas de autorização.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MSKReplicatorIAMPassRole",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/MSKReplicationRole",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "kafka.amazonaws.com"
        }
      }
    },
    {
      "Sid": "MSKReplicatorServiceLinkedRole",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::123456789012:role/aws-service-role/
kafka.amazonaws.com/AWSServiceRoleForKafka*"
```

```
    },
    {
      "Sid": "MSKReplicatorEC2Actions",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVpcs",
        "ec2:CreateNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:123456789012:subnet/subnet-0abcd1234ef56789",
        "arn:aws:ec2:us-east-1:123456789012:security-group/sg-0123abcd4567ef89",
        "arn:aws:ec2:us-east-1:123456789012:network-
interface/eni-0a1b2c3d4e5f67890",
        "arn:aws:ec2:us-east-1:123456789012:vpc/vpc-0a1b2c3d4e5f67890"
      ]
    },
    {
      "Sid": "MSKReplicatorActions",
      "Effect": "Allow",
      "Action": [
        "kafka:CreateReplicator",
        "kafka:TagResource"
      ],
      "Resource": [
        "arn:aws:kafka:us-
east-1:123456789012:cluster/myCluster/abcd1234-56ef-78gh-90ij-klmnopqrstuv",
        "arn:aws:kafka:us-
east-1:123456789012:replicator/myReplicator/wxyz9876-54vu-32ts-10rq-ponmlkjihgfe"
      ]
    }
  ]
}
```

Veja a seguir um exemplo de política do IAM para descrever o replicador. É necessário usar a ação `kafka:DescribeReplicator` ou a ação `kafka:ListTagsForResource`, mas não ambas.

JSON

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "kafka:DescribeReplicator",
                "kafka:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

Tipos e versões de clusters compatíveis para o Replicador do MSK

Estes são os requisitos para tipos de instância, versões do Kafka e configurações de rede compatíveis.

- O replicador do MSK oferece suporte a qualquer combinação de clusters provisionados do MSK e clusters do MSK com tecnologia sem servidor como clusters de origem e destino. No momento, o replicador do MSK não é compatível com outros tipos de clusters do Kafka.
- Os clusters do MSK com a tecnologia sem servidor exigem controle de acesso do IAM, não oferecem suporte à replicação de ACL do Apache Kafka e têm compatibilidade limitada com a replicação de configuração no tópico. Consulte O que é o MSK Sem Servidor?.
- O MSK Replicator é suportado somente em clusters que executam o Apache Kafka 2.7.0 ou superior, independentemente de seus clusters de origem e de destino estarem no mesmo ou em diferentes. Regiões da AWS
- O MSK Replicator oferece suporte a clusters usando tipos de instância m5.large ou maiores. Não há suporte para clusters t3.small.
- Se você estiver usando o replicador do MSK com um cluster provisionado pelo MSK, precisará de, no mínimo, três agentes nos clusters de origem e de destino. É possível replicar dados entre clusters em duas zonas de disponibilidade, mas você precisaria de um mínimo de quatro agentes nesses clusters.
- Os clusters MSK de origem e de destino devem estar na mesma AWS conta. Não há compatibilidade com a replicação entre clusters em contas diferentes.

- Se os clusters MSK de origem e de destino estiverem em AWS regiões diferentes (entre regiões), o MSK Replicator exigirá que o cluster de origem tenha a conectividade privada de várias VPCs ativada para seu método de controle de acesso IAM.

  Várias VPCs não são necessárias para outros métodos de autenticação no cluster de origem para a replicação do MSK. Regiões da AWS

  Várias VPCs também não são necessárias se você estiver replicando dados entre clusters no mesmo. Região da AWS Consulte the section called “Conectividade privada multi-VPC em uma única região”.
- A replicação de nomes de tópicos idênticos (mantenha o mesmo nome de tópicos no console) requer um cluster do MSK executando o Kafka versão 2.8.1 ou superior.
- Para configurações de replicação de nomes de tópicos idênticos (mantenha o mesmo nome de tópicos no console), para evitar o risco de replicação cíclica, não faça alterações nos cabeçalhos que o Replicador do MSK cria (`__mskmr`).

Configuração do cluster compatível do MSK Sem Servidor

- O MSK Serverless é compatível com a replicação destas configurações de tópicos para clusters de destino do MSK Serverless durante a criação do tópico: `cleanup.policy`, `compression.type`, `max.message.bytes`, `retention.bytes`, `retention.ms`.
- O MSK Serverless só é compatível com estas configurações de tópicos durante a sincronização da configuração de tópicos: `compression.type`, `max.message.bytes`, `retention.bytes`, `retention.ms`.
- O replicador usa 83 partições compactadas nos clusters de destino do MSK Serverless. Certifique-se de que os clusters de destino do MSK Serverless tenham um número suficiente de partições compactadas. Consulte Cota do MSK Serverless.

Alterações na configuração de cluster

- Recomenda-se que você não ative ou desative o armazenamento em camadas após a criação do replicador do MSK. Se seu cluster de destino não estiver em camadas, o MSK não copiará as configurações de armazenamento em camadas, independentemente de seu cluster de origem estar ou não com essa configuração. Se você ativar o armazenamento em camadas no cluster de destino após a criação do replicador, será necessário recriar o replicador. Se você quiser copiar dados de um cluster que não esteja em camadas para um cluster em camadas, você não deve

copiar as configurações de tópico. Consulte Habilitar e desabilitar o armazenamento em camadas em um tópico existente.

- Não altere as configurações do cluster após a criação do replicador do MSK. As configurações do cluster são validadas durante a criação do replicador do MSK. Para evitar problemas com o replicador do MSK, não altere as configurações a seguir após a criação do replicador do MSK.
  - Altere o cluster do MSK para o tipo de instância t3.
  - Altere as permissões do perfil de execução do serviço.
  - Desabilite a conectividade privada multi-VPC do MSK.
  - Altere a política baseada em recursos anexada do cluster.
  - Altere as regras de grupos de segurança de cluster.

Crie um replicador usando o console da AWS na região do cluster de destino

A seção a seguir explica o fluxo de trabalho do console por etapas para criar um replicador.

Detalhes do replicador

1. Na AWS região em que seu cluster MSK de destino está localizado, abra o console do Amazon MSK em casahttps://console.aws.amazon.com/msk/? region=us-east-1#/home/.
2. Escolha Replicadores para exibir a lista de replicadores na conta.
3. Escolha Criar replicador.
4. No painel Detalhes do replicador, dê um nome exclusivo ao novo replicador.

Escolher seu cluster de origem

O cluster de origem contém os dados que você deseja copiar para um cluster de destino do MSK.

1. No painel Cluster de origem, escolha a região da AWS do cluster de origem.

   Você pode consultar a região de um cluster acessando Clusters do MSK e examinando o ARN dos detalhes do cluster. O nome da região está incorporado na string do ARN. No exemplo de ARN a seguir, `ap-southeast-2` é a região do cluster.

```
arn:aws:kafka:ap-southeast-2:123456789012:cluster/cluster-11/
eec93c7f-4e8b-4baf-89fb-95de01ee639c-s1
```

2. Insira o ARN do seu cluster de origem ou navegue para escolher seu cluster de origem.
3. Escolha uma ou mais sub-redes para seu cluster de origem.

   O console exibe as sub-redes disponíveis na região do cluster de origem para você selecionar. Você deve selecionar, no mínimo, duas sub-redes. Para um replicador do MSK na mesma região, as sub-redes que você seleciona para acessar o cluster de origem e as sub-redes para acessar o cluster de destino devem estar na mesma zona de disponibilidade.
4. Escolha grupos de segurança para que o Replicador do MSK acesse o cluster de origem.

   - Para replicação entre regiões (CRR), você não precisa fornecer grupos de segurança para o cluster de origem.
   - Para replicação na mesma região (SRR), acesse o EC2 console da Amazon em https://console.aws.amazon.com/ec2/ e certifique-se de que os grupos de segurança que você fornecerá para o Replicador tenham regras de saída para permitir o tráfego para os grupos de segurança do seu cluster de origem. Além disso, certifique-se de que os grupos de segurança do cluster de origem tenham regras de saída que permitam o tráfego para os grupos de segurança do replicador fornecidos para a origem.

     Para adicionar regras de entrada ao seu grupo de segurança do cluster de origem:

     1. No AWS console, acesse os detalhes do cluster de origem selecionando o nome do cluster.
     2. Selecione a guia Propriedades e, em seguida, role para baixo até o painel Configurações de rede para selecionar o nome do Grupo de segurança aplicado.
     3. Acesse as regras de entrada e selecione Editar regras de entrada.
     4. Selecione Adicionar regra.
     5. Na coluna Tipo para a nova regra, selecione TCP personalizado.
     6. Na coluna Intervalo de portas, digite 9098. O Replicador do MSK usa o controle de acesso do IAM para se conectar ao cluster que usa a porta 9098.
     7. Na coluna Origem, digite o nome do grupo de segurança que você fornecerá durante a criação do replicador para o cluster de origem (pode ser igual ao grupo de segurança do cluster de origem do MSK) e selecione Salvar regras.

Para adicionar regras de saída ao grupo de segurança do replicador fornecido para a origem:

1. No AWS console da Amazon EC2, acesse o grupo de segurança que você fornecerá durante a criação do Replicator para a fonte.
2. Acesse as regras de saída e selecione Editar regras de saída.
3. Selecione Adicionar regra.
4. Na coluna Tipo para a nova regra, selecione TCP personalizado.
5. Na coluna Intervalo de portas, digite `9098`. O Replicador do MSK usa o controle de acesso do IAM para se conectar ao cluster que usa a porta 9098.
6. Na coluna Origem, digite o nome do grupo de segurança do cluster de origem do MSK e selecione Salvar regras.

> **Note**
>
> Como alternativa, caso não queira restringir o tráfego usando os grupos de segurança, você poderá adicionar regras de entrada e saída que permitam todo o tráfego.
> 1. Selecione Adicionar regra.
> 2. Na coluna Tipo, escolha Todo o tráfego.
> 3. Na coluna Origem, digite `0.0.0.0/0` e selecione Salvar regras.

Escolher seu cluster de destino

O cluster de destino é o cluster do MSK provisionado ou com tecnologia sem servidor para o qual os dados de origem são copiados.

> **Note**
>
> O replicador do MSK cria novos tópicos no cluster de destino com um prefixo gerado automaticamente adicionado ao nome do tópico. Por exemplo, o replicador do MSK replica dados em "`topic`" com base no cluster de origem para um novo tópico chamado `<sourceKafkaClusterAlias>.topic` no cluster de destino. Isso serve para distinguir entre tópicos que contenham dados replicados do cluster de origem de outros tópicos no cluster de destino e para evitar que os dados sejam replicados circularmente entre os

clusters. Você pode encontrar o prefixo que será adicionado aos nomes dos tópicos no cluster de destino no campo sourceKafkaClusterAlias usando a `DescribeReplicator` API ou a página de detalhes do Replicator no console MSK. O prefixo no cluster de destino é `<sourceKafkaClusterAlias>`.

1. No painel Cluster de destino, escolha a AWS região em que o cluster de destino está localizado.
2. Insira o ARN do seu cluster de destino ou navegue para escolher seu cluster de destino.
3. Escolha uma ou mais sub-redes para seu cluster de destino.

   O console exibe as sub-redes disponíveis na região do cluster de destino para você selecionar. Selecione ao menos duas sub-redes.
4. Escolha grupos de segurança para que o Replicador do MSK acesse o cluster de destino.

   Os grupos de segurança disponíveis na região do cluster de destino são exibidos para você selecionar. O grupo de segurança escolhido será associado a cada conexão. Para obter mais informações sobre o uso de grupos de segurança, consulte Controle o tráfego para seus AWS recursos usando grupos de segurança no Guia do usuário da Amazon VPC.

   - Tanto para replicação entre regiões (CRR) quanto para replicação na mesma região (SRR), acesse o EC2 console da Amazon em https://console.aws.amazon.com/ec2/e certifique-se de que os grupos de segurança que você fornecerá ao Replicador tenham regras de saída para permitir o tráfego para os grupos de segurança do seu cluster de destino. Além disso, certifique-se de que os grupos de segurança do seu cluster de destino tenham regras de entrada que aceitem o tráfego proveniente dos grupos de segurança do replicador fornecidos para o destino.

     Para adicionar regras de entrada ao grupo de segurança do cluster de destino:

     1. No AWS console, acesse os detalhes do cluster de destino selecionando o nome do cluster.
     2. Selecione a guia Propriedades e, em seguida, role para baixo até o painel Configurações de rede para selecionar o nome do Grupo de segurança aplicado.
     3. Acesse as regras de entrada e selecione Editar regras de entrada.
     4. Selecione Adicionar regra.
     5. Na coluna Tipo para a nova regra, selecione TCP personalizado.

6. Na coluna Intervalo de portas, digite 9098. O Replicador do MSK usa o controle de acesso do IAM para se conectar ao cluster que usa a porta 9098.
7. Na coluna Origem, digite o nome do grupo de segurança que você fornecerá durante a criação do replicador para o cluster de destino (pode ser igual ao grupo de segurança do cluster de destino do MSK) e selecione Salvar regras.

Para adicionar regras de saída ao grupo de segurança do replicador fornecido para o destino:

1. No AWS console, acesse o grupo de segurança que você fornecerá durante a criação do Replicator para o destino.
2. Selecione a guia Propriedades e, em seguida, role para baixo até o painel Configurações de rede para selecionar o nome do Grupo de segurança aplicado.
3. Acesse as regras de saída e selecione Editar regras de saída.
4. Selecione Adicionar regra.
5. Na coluna Tipo para a nova regra, selecione TCP personalizado.
6. Na coluna Intervalo de portas, digite 9098. O Replicador do MSK usa o controle de acesso do IAM para se conectar ao cluster que usa a porta 9098.
7. Na coluna Origem, digite o nome do grupo de segurança do cluster de destino do MSK e selecione Salvar regras.

> **Note**
>
> Como alternativa, caso não queira restringir o tráfego usando os grupos de segurança, você poderá adicionar regras de entrada e saída que permitam todo o tráfego.
> 1. Selecione Adicionar regra.
> 2. Na coluna Tipo, escolha Todo o tráfego.
> 3. Na coluna Origem, digite `0.0.0.0/0` e selecione Salvar regras.

Definir configurações e permissões do replicador

1. No painel Configurações do replicador, especifique os tópicos que deseja replicar usando expressões regulares nas listas de permissão e proibição. Todos os tópicos são replicados por padrão.

> **Note**
>
> O Replicador do MSK replica somente até 750 tópicos de forma ordenada. Se você precisar replicar mais tópicos, recomendamos criar um replicador separado. Acesse o Support Center do AWS console e crie um caso de suporte se precisar de suporte para mais de 750 tópicos por replicador. Você pode monitorar o número de tópicos que estão sendo replicados usando a métrica TopicCount "". Consulte Cota de corretor Amazon MSK Standard.

2. Por padrão, o Replicador do MSK inicia a replicação a partir do último (mais recente) deslocamento nos tópicos selecionados. Como alternativa, você pode iniciar a replicação a partir do primeiro (mais antigo) deslocamento nos tópicos selecionados se quiser replicar os dados existentes em seus tópicos. Depois que o replicador for criado, você não poderá alterar essa configuração. Essa configuração corresponde ao `startingPosition`campo na `CreateReplicator`solicitação e na `DescribeReplicator`resposta APIs.
3. Escolha uma configuração de nome de tópico:

   - Replicação do nome do tópico `PREFIXED` (Adicionar prefixo ao nome dos tópicos no console): a configuração padrão. O Replicador do MSK replica “topic1” do cluster de origem para um novo tópico no cluster de destino denominado `<sourceKafkaClusterAlias>.topic1`.
   - Replicação de nomes do tópicos idênticos (mantenha o mesmo nome de tópicos no console): os tópicos do cluster de origem são replicados com nomes de tópicos idênticos no cluster de destino.

   Essa configuração corresponde ao `TopicNameConfiguration` campo na `CreateReplicator` solicitação e na `DescribeReplicator` resposta APIs. Consulte Funcionamento do replicador do Amazon MSK.

   > **Note**
   >
   > Por padrão, o Replicador do MSK cria tópicos no cluster de destino com um prefixo gerado automaticamente adicionado ao nome do tópico. Isso serve para distinguir entre tópicos que contenham dados replicados do cluster de origem de outros tópicos no cluster de destino e para evitar que os dados sejam replicados circularmente entre os clusters. Como alternativa, você pode criar um Replicador do MSK com replicação de nomes de tópicos idênticos (mantenha o mesmo nome de tópicos no console) para que

os nomes dos tópicos sejam preservados durante a replicação. Essa configuração reduz a necessidade de reconfigurar aplicações cliente durante a configuração e simplifica a operação de arquiteturas de streaming de vários clusters.

4. Por padrão, o MSK Replicator copia todos os metadados, incluindo configurações de tópicos, listas de controle de acesso (ACLs) e compensações de grupos de consumidores para um failover contínuo. Se você não estiver criando o replicador para failover, é possível optar por desativar uma ou mais dessas configurações disponíveis na seção Configurações adicionais.

> **Note**
>
> O MSK Replicator não replica a gravação, ACLs pois seus produtores não devem escrever diretamente no tópico replicado no cluster de destino. Seus produtores devem gravar no tópico local no cluster de destino após o failover. Para mais detalhes, consulte Execute um failover planejado para a região secundária AWS.

5. No painel Replicação do grupo de consumidores, especifique os grupos de consumidores que deseja replicar usando expressões regulares nas listas de permissão e proibição. Todos os grupos de consumidores são replicados por padrão.
6. No painel Compactação, você pode optar opcionalmente por compactar os dados gravados no cluster de destino. Se você for usar a compactação, recomendamos que use o mesmo método de compactação dos dados em seu cluster de origem.
7. No painel Permissões de acesso, execute uma das seguintes ações:

   a. Selecione Criar ou atualizar o perfil do IAM com as políticas necessárias. O console do MSK anexará automaticamente as permissões e a política de confiança necessárias ao perfil de execução do serviço necessário para ler e gravar em seus clusters de origem e destino do MSK.

   b. Forneça seu próprio perfil do IAM selecionando Escolher entre os perfis do IAM que o Amazon MSK pode assumir. Recomendamos que você anexe a política `AWSMSKReplicatorExecutionRole` gerenciada do IAM ao perfil de execução do serviço, em vez de escrever sua própria política do IAM.

      - Crie o perfil do IAM que o replicador usará para ler e gravar em seus clusters de origem e destino do MSK com o JSON abaixo como parte da política de confiança e

o `AWSMSKReplicatorExecutionRole` anexado ao perfil. Na política de confiança, substitua o espaço reservado <yourAccountID> pelo ID efetivo da sua conta.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "kafka.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<yourAccountID>"
                }
            }
        }
    ]
}
```

8. No painel Tags do replicador, você pode, opcionalmente, atribuir tags ao recurso replicador do MSK. Para obter mais informações, consulte Marcar um cluster do Amazon MSK. Para um replicador do MSK entre regiões, as tags são sincronizadas automaticamente com a região remota quando o replicador é criado. Se você alterar as tags após a criação do replicador, a alteração não será sincronizada automaticamente com a região remota, então você precisará sincronizar manualmente as referências do replicador local e do replicador remoto.
9. Escolha Criar.

Se você quiser restringir a permissão `kafka-cluster:WriteData`, consulte a seção Criar políticas de autorização de Como funciona o controle de acesso do IAM para o Amazon MSK. Você precisará adicionar a permissão `kafka-cluster:WriteDataIdempotently` ao cluster de origem e de destino.

A criação e transferência do replicador do MSK para o status RUNNING leva aproximadamente 30 minutos.

Se você criar um novo replicador do MSK para substituir um que você excluiu, o novo replicador iniciará a replicação a partir do último deslocamento.

Se o replicador do MSK tiver passado para o status FAILED, consulte a seção Solução de problemas do replicador do MSK.

Editar configurações do replicador do MSK

Você não pode alterar o cluster de origem, o cluster de destino, a posição inicial do replicador ou a configuração de replicação do nome do tópico após o Replicador do MSK ter sido criado. Você precisa criar um novo replicador para usar a configuração de replicação de nomes de tópicos idênticos. No entanto, você pode editar outras configurações do replicador, como tópicos e grupos de consumidores a replicar.

1. Faça login no AWS Management Console e abra o console Amazon MSK em https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.
2. No painel de navegação esquerdo, escolha Replicadores para exibir a lista de replicadores na conta e selecione o Replicador do MSK que deseja editar.
3. Escolha a guia Properties (Propriedades).
4. Na seção Configurações do replicador, escolha Editar replicador.
5. Você pode editar as configurações do replicador do MSK alterando qualquer uma dessas configurações.

    - Especifique os tópicos que deseja replicar usando expressões regulares nas listas de permissão e proibição. Por padrão, o MSK Replicator copia todos os metadados, incluindo configurações de tópicos, listas de controle de acesso (ACLs) e compensações de grupos de consumidores para um failover contínuo. Se você não estiver criando o replicador para failover, é possível optar por desativar uma ou mais dessas configurações disponíveis na seção Configurações adicionais.

    > **Note**
    >
    > O MSK Replicator não replica a gravação, ACLs pois seus produtores não devem escrever diretamente no tópico replicado no cluster de destino. Seus produtores devem gravar no tópico local no cluster de destino após o failover. Para mais detalhes, consulte Execute um failover planejado para a região secundária AWS.

- Em Replicação do grupo de consumidores, é possível especificar os grupos de consumidores que deseja replicar usando expressões regulares nas listas de permissão e proibição. Todos os grupos de consumidores são replicados por padrão. Se as listas de permissão e proibição estiverem vazias, a replicação do grupo de consumidores será desativada.
- No painel Tipo de compactação do destino, você pode optar por compactar ou não os dados gravados no cluster de destino. Se você for usar a compactação, recomendamos que use o mesmo método de compactação dos dados em seu cluster de origem.

6. Salve as alterações.

   A criação e transferência do replicador do MSK para o estado em execução leva aproximadamente 30 minutos. Se o replicador do MSK tiver passado para o status FAILED, consulte a seção de solução de problemas ???.

Excluir um replicador do MSK

Talvez seja necessário excluir um replicador do MSK se ele falhar na criação (status FAILED). Os clusters de origem e destino atribuídos a um replicador do MSK não podem ser alterados após a criação do replicador do MSK. Você pode excluir um replicador do MSK existente e criar um novo. Se você criar um novo replicador do MSK para substituir o excluído, o novo replicador iniciará a replicação com base na última compensação.

1. Na AWS região em que seu cluster de origem está localizado, faça login e abra o AWS Management Console console do Amazon MSK em https://console.aws.amazon.com/msk/casa?region=us-east-1#/home/.
2. No painel de navegação, selecione Replicadores.
3. Na lista de replicadores do MSK, selecione o que você deseja excluir e escolha Excluir.

Monitorar a replicação

Você pode usar https://console.aws.amazon.com/cloudwatch/na região do cluster de destino para visualizar métricas de`ReplicationLatency`,`MessageLag`, e `ReplicatorThroughput` em um nível de tópico e agregado para cada Amazon MSK Replicator. As métricas são visíveis abaixo ReplicatorNameno namespace “AWS/Kafka”. Você também pode ver as métricas `ReplicatorFailure`, `AuthError` e `ThrottleTime` para verificar se há problemas.

O console MSK exibe um subconjunto de CloudWatch métricas para cada replicador MSK. Na lista Replicadores do console, selecione o nome de um replicador e selecione a guia Monitoramento.

Métricas de replicador do MSK

As métricas a seguir descrevem as métricas de desempenho ou conexão do replicador do MSK.

AuthError as métricas não abrangem erros de autenticação em nível de tópico. Para monitorar os erros de autenticação em nível de tópico do MSK Replicator, monitore as métricas do Replicator e as ReplicationLatency métricas em nível de tópico do cluster de origem,. MessagesInPerSec Se um tópico ReplicationLatency cair para 0, mas o tópico ainda tiver dados sendo produzidos, isso indica que o replicador tem um problema de autenticação com o tópico. Verifique se o perfil do IAM de execução do serviço do replicador tem permissão suficiente para acessar o tópico.

Tipo de métrica	Métrica	Descrição	Dimensõ	Unidade	Granular dade métrica bruta	Estatísti ca bruta de agregaçã métrica	
Performa ce	Replicati onLatency	O tempo necessário para que os registros sejam replicado s da origem para o cluster de destino; a duração entre o tempo de produção do registro na origem e o tempo de replicação no destino. Se Replicati onLatency	Replicato rName	Milisegu ndos	Partition	Máximo	
			Replicato rName, Tópico	Milisegu ndos	Partition	Máximo	

Tipo de métrica	Métrica	Descrição	Dimensõ	Unidade	Granularidade métrica bruta	Estatística bruta de agregaçã métrica	
		aumentar, verifique se os clusters têm partições suficientes para suportar a replicação. Pode ocorrer alta latência de replicação quando a contagem de partições for muito baixa para um throughput alto.					

Tipo de métrica	Métrica	Descrição	Dimensõ	Unidade	Granularidade métrica bruta	Estatística bruta de agregaçã métrica
Performance	MessageLag	Monitora a sincronização entre o MSK Replicator e o cluster de origem. MessageLag indica o atraso entre as mensagens produzidas no cluster de origem e as mensagens consumidas pelo replicador. Não é o atraso entre o cluster de origem e o de destino. Mesmo que o cluster de origem esteja indisponível ou interrompido, o replicador terminará de gravar a mensagem consumida no cluster de destino. Depois	ReplicatorName	Contagem	Partition	Soma
			ReplicatorName, Tópico	Contagem	Partition	Soma

Tipo de métrica	Métrica	Descrição	Dimensõ	Unidade	Granularidade métrica bruta	Estatística bruta de agregaçã métrica	
		de uma interrupção, MessageLag mostra um aumento indicando o número de mensagens que o replicador está por trás do cluster de origem e isso pode ser monitorado até que o número de mensagens seja 0, mostrando que o replicador alcançou o cluster de origem.					

Tipo de métrica	Métrica	Descrição	Dimensõ	Unidade	Granular dade métrica bruta	Estatísti ca bruta de agregaçã métrica
Performa ce	Replicato rBytesInPerSec	Número médio de bytes processados pelo replicado r por segundo. Os dados processados pelo Replicado r do MSK consistem em todos os dados que o Replicador do MSK recebe, incluindo os dados replicado s para o cluster de destino e os dados filtrados pelo Replicador do MSK (somente se o replicador estiver configura do com uma configuração de nomes de tópicos idênticos) para evitar que os dados	Replicato rName	BytesPer econd	Replicato rName	Soma

Tipo de métrica	Métrica	Descrição	Dimensõ	Unidade	Granularidade métrica bruta	Estatística bruta de agregaçã métrica	
		sejam copiados de volta para o mesmo tópico de origem. Se o replicador estiver configurado com a configuração de nome de tópico “Prefixado”, ambas as métricas `ReplicatorBytesInPerSec` e `ReplicatorThroughput` terão o mesmo valor, pois nenhum dado será filtrado pelo Replicador do MSK.					

Tipo de métrica	Métrica	Descrição	Dimensõ	Unidade	Granular dade métrica bruta	Estatísti ca bruta de agregaçã métrica
Performa ce	Replicato rThroughput	Número médio de bytes replicados por segundo. Se ReplicatorThroughput optar por um tópico, verifique KafkaClusterPingSuccessCount AuthError as métricas para garantir que o replicador possa se comunicar com os clusters, verifique as métricas do cluster para garantir que o cluster não esteja inativo.	ReplicatorName	BytesPer econd	Partition	Soma
			ReplicatorName, Tópico	BytesPer econd	Partition	Soma

Tipo de métrica	Métrica	Descrição	Dimensõ	Unidade	Granular dade métrica bruta	Estatísti ca bruta de agregaçã métrica
Depure	AuthError	O número de conexões com falha na autenticação por segundo. Se essa métrica estiver acima de 0, você poderá verificar se a política do perfil de execução do serviço para o replicado r é válida e garantir que não haja recusa de permissões definidas para as permissõe s do cluster. Com base na dimensão ClusterAlias, você pode identificar se o cluster de origem ou de destino está apresenta	Replicato rName, ClusterA ias	Contage	Operado	Soma

Tipo de métrica	Métrica	Descrição	Dimensõ	Unidade	Granularidade métrica bruta	Estatística bruta de agregaçã métrica	
		ndo erros de autenticação.					

Tipo de métrica	Métrica	Descrição	Dimensõ	Unidade	Granular dade métrica bruta	Estatísti ca bruta de agregaçã métrica
Depure	ThrottleTime	O tempo médio em ms em que uma solicitaç ão passou por controle de utilização pelos agentes no cluster. Defina o controle de utilização para evitar que o replicador do MSK sobrecarr egue o cluster. Se essa métrica for 0, a latência de replicaçã o não for alta e o replicato rThroughput for o esperado, o controle de utilização estará funcionando conforme o esperado. Se essa métrica estiver acima de 0, você poderá	Replicato rName, ClusterA ias	Milisegu ndos	Operado	Máximo

Tipo de métrica	Métrica	Descrição	Dimensõ	Unidade	Granularidade métrica bruta	Estatística bruta de agregaçã métrica	
		ajustar o controle de utilização adequadamente.					
Depure	ReplicatorFailure	O número de falhas que o replicador está enfrentando.	Replicato rName	Contage		Soma	

Tipo de métrica	Métrica	Descrição	Dimensõ	Unidade	Granular dade métrica bruta	Estatísti ca bruta de agregaçã métrica
Depure	KafkaClus terPingSu ccessCount	Indica a integrida de da conexão do replicador com o cluster do Kafka. Se esse valor for 1, a conexão está íntegra. Se o valor for 0 ou não houver nenhum ponto de dados, a conexão não está íntegra. Se o valor for 0, você poderá verificar as configurações de permissão de rede ou IAM para o cluster do Kafka. Com base na ClusterAlias dimensão, você pode identificar se essa métrica é para o cluster de origem ou de destino.	Replicato rName, ClusterA ias	Contage		Soma

Usar a replicação para aumentar a resiliência de uma aplicação de streaming do Kafka em todas as regiões

Você pode usar o MSK Replicator para configurar topologias de cluster ativo-ativo ou ativo-passivo para aumentar a resiliência do seu aplicativo Apache Kafka em todas as regiões. AWS Em uma configuração ativa-ativa, os dois clusters do MSK estão atendendo ativamente leituras e gravações. Em uma configuração ativa-passiva, somente um cluster do MSK por vez estará atendendo ativamente dados de streaming, enquanto o outro cluster estará em espera.

Considerações para criar aplicações do Apache Kafka em várias regiões

Seus consumidores devem ser capazes de reprocessar mensagens duplicadas sem impacto posterior. O MSK Replicator replica dados at-least-once que podem resultar em duplicatas no cluster em espera. Quando você muda para a AWS região secundária, seus consumidores podem processar os mesmos dados mais de uma vez. O replicador do MSK prioriza a cópia de dados em vez das compensações do consumidor para melhorar o desempenho. Após um failover, o consumidor pode começar a ler as compensações anteriores, resultando em processamento duplicado.

Produtores e consumidores também devem tolerar a perda mínima de dados. Como o MSK Replicator replica dados de forma assíncrona, quando a AWS região primária começa a apresentar falhas, não há garantia de que todos os dados sejam replicados para a região secundária. Você pode usar a latência de replicação para determinar o máximo de dados que não foram copiados para a região secundária.

Uso da topologia ativa-ativa vs. ativa-passiva de cluster

Uma topologia ativa-ativa de cluster oferece quase zero tempo de recuperação e a capacidade de sua aplicação de streaming operar simultaneamente em várias regiões da AWS . Quando um cluster em uma região está comprometido, as aplicações conectadas ao cluster na outra região continuam processando dados.

As configurações ativa-passiva são adequadas para aplicações que podem ser executadas em apenas uma região da AWS por vez ou quando você precisa de mais controle sobre a ordem de processamento de dados. As configurações ativa-passiva exigem mais tempo de recuperação do que as configurações ativa-ativa, pois você deve iniciar toda a configuração ativa-passiva, incluindo seus produtores e consumidores, na região secundária para retomar o streaming de dados após um failover.

Criar uma configuração ativa-passiva de cluster do Kafka com as configurações de nomenclatura de tópicos recomendadas

Para uma configuração ativa-passiva, recomendamos que você opere uma configuração semelhante de produtores, clusters MSK e consumidores (com o mesmo nome de grupo de consumidores) em duas regiões diferentes. AWS É importante que os dois clusters do MSK tenham capacidade idêntica de leitura e gravação para garantir a replicação confiável dos dados. Você precisa criar um replicador do MSK para copiar continuamente os dados do cluster primário para o cluster em espera. Você também precisa configurar seus produtores para gravar dados em tópicos em um cluster na mesma AWS região.

Para uma configuração ativa-passiva, crie um replicador com replicação de nomes de tópicos idênticos (mantenha o mesmo nome de tópicos no console) para começar a replicar dados do cluster do MSK na região primária para o cluster na região secundária. Recomendamos que você opere um conjunto duplicado de produtores e consumidores nas duas AWS regiões, cada um se conectando ao cluster em sua própria região usando sua string de bootstrap. Isso simplifica o processo de failover, pois não exigirá alterações na string de bootstrap. Para garantir que os consumidores leiam próximo de onde pararam, os consumidores nos clusters de origem e de destino devem ter o mesmo ID de grupo de consumidores.

Se você usar a replicação de nomes de tópicos indênticos (mantenha o mesmo nome de tópicos no console) para o Replicador do MSK, ele replicará os tópicos com o mesmo nome dos tópicos de origem correspondentes.

Recomendamos que você defina as configurações e permissões no nível de cluster para os clientes no cluster de destino. Você não precisa definir as configurações de nível de tópico e a leitura literal, ACLs pois o MSK Replicator as copia automaticamente se você tiver selecionado a opção de copiar listas de controle de acesso. Consulte Replicação de metadados.

Failover para a região secundária da AWS

Recomendamos que você monitore a latência da replicação na AWS região secundária usando a Amazon. CloudWatch Durante um evento de serviço na AWS região principal, a latência da replicação pode aumentar repentinamente. Se a latência continuar aumentando, use o AWS Service Health Dashboard para verificar se há eventos de serviço na AWS região principal. Se houver um evento, você pode fazer o failover para a AWS região secundária.

Execute um failover planejado para a região secundária AWS

Você pode realizar um failover planejado para testar a resiliência do seu aplicativo contra um evento inesperado em sua AWS região primária, que tem seu cluster MSK de origem. Um failover planejado não deve resultar em perda de dados.

Se você estiver usando a configuração de replicação de nomes de tópicos idênticos, siga estas etapas:

1. Desligue todos os produtores e consumidores que se conectam ao seu cluster de origem.
2. Crie um Replicador do MSK para replicar dados do cluster do MSK na região secundária para o cluster do MSK na região primári com replicação de nomes de tópicos idênticos (mantenha o mesmo nome de tópicos no console). Isso é necessário para copiar os dados que você gravará na região secundária de volta para a região primária, para que você possa fazer failback para a região primária após o término do evento inesperado.
3. Inicie produtores e consumidores conectados ao cluster de destino na AWS região secundária.

Se você estiver usando a configuração de nomes de tópicos prefixados, siga estas etapas para fazer o failover:

1. Desligue todos os produtores e consumidores que se conectam ao seu cluster de origem.
2. Crie um novo replicador do MSK para replicar dados do seu cluster do MSK na região secundária para o seu cluster do MSK na região primária. Isso é necessário para copiar os dados que você gravará na região secundária de volta para a região primária, para que você possa fazer failback para a região primária após o término do evento inesperado.
3. Inicie produtores no cluster de destino na AWS região secundária.
4. Dependendo dos requisitos de ordenação de mensagens da aplicação, siga as etapas em uma das guias a seguir.

   No message ordering

   Se seu aplicativo não exigir a ordenação de mensagens, inicie consumidores na AWS região secundária que leiam os tópicos locais (por exemplo, tópico) e replicados (por exemplo,`<sourceKafkaClusterAlias>.topic`) usando um operador curinga (por exemplo,). `.*topic`

Message ordering

Se sua aplicação exigir a ordenação de mensagens, inicie os consumidores somente para os tópicos replicados no cluster de destino (p. ex., `<sourceKafkaClusterAlias>.topic`), mas não para os tópicos locais (p. ex., `topic`).

5. Aguarde até que todos os consumidores de tópicos replicados no cluster de destino do MSK concluam o processamento de todos os dados, para que o atraso do consumidor seja 0 e o número de registros processados também seja 0. Em seguida, interrompa os consumidores dos tópicos replicados no cluster de destino. Nesse ponto, todos os registros que foram replicados do cluster do MSK de origem para o cluster do MSK de destino foram consumidos.
6. Inicie consumidores para os tópicos locais (p. ex., `topic`) no cluster de destino do MSK.

Execute um failover não planejado para a região secundária AWS

Você pode realizar um failover não planejado quando há um evento de serviço na AWS região primária que tem seu cluster MSK de origem e você deseja redirecionar temporariamente seu tráfego para a região secundária que tem seu cluster MSK de destino. Um failover não planejado pode resultar na perda de alguns dados, pois o Replicador do MSK replica dados de modo assíncrono. Você pode monitorar o atraso da mensagem usando as métricas em ???.

Se você estiver usando uma configuração de replicação de nomes de tópicos idênticos (mantenha o mesmo nome de tópicos no console), siga estas etapas:

1. Tente desligar todos os produtores e consumidores que se conectam ao cluster de origem do MSK na região primária. Essa operação pode não ter êxito devido a deficiências na região.
2. Faça com que produtores e consumidores se conectem ao cluster MSK de destino na AWS região secundária para concluir o failover. Como o MSK Replicator também replica metadados, incluindo compensações de leitura ACLs e de grupos de consumidores, seus produtores e consumidores retomarão o processamento sem problemas de onde pararam antes do failover.

Se você estiver usando a configuração de nomes de tópicos `PREFIX`, siga estas etapas para fazer o failover:

1. Tente desligar todos os produtores e consumidores que se conectam ao cluster de origem do MSK na região primária. Essa operação pode não ter êxito devido a deficiências na região.

2. Faça com que produtores e consumidores se conectem ao cluster MSK de destino na AWS região secundária para concluir o failover. Como o MSK Replicator também replica metadados, incluindo compensações de leitura ACLs e de grupos de consumidores, seus produtores e consumidores retomarão o processamento sem problemas de onde pararam antes do failover.
3. Dependendo dos requisitos de ordenação de mensagens da aplicação, siga as etapas em uma das guias a seguir.

   No message ordering

   Se seu aplicativo não exigir a ordenação de mensagens, inicie consumidores na AWS região de destino que leiam os tópicos locais (por exemplo`topic`) e replicados (por exemplo,`<sourceKafkaClusterAlias>.topic`) usando um operador curinga (por exemplo,). `.*topic`

   Message ordering

   1. Inicie os consumidores somente para os tópicos replicados no cluster de destino (p. ex., `<sourceKafkaClusterAlias>.topic`), mas não para os tópicos locais (p. ex., `topic`).
   2. Aguarde até que todos os consumidores de tópicos replicados no cluster de destino do MSK concluam o processamento de todos os dados, para que o atraso do deslocamento seja 0 e o número de registros processados também seja 0. Em seguida, interrompa os consumidores dos tópicos replicados no cluster de destino. Nesse ponto, todos os registros que foram replicados do cluster do MSK de origem para o cluster do MSK de destino foram consumidos.
   3. Inicie consumidores para os tópicos locais (p. ex., `topic`) no cluster de destino do MSK.
4. Depois que o evento de serviço terminar na região primária, crie um Replicador do MSK para replicar dados do cluster do MSK na região secundária para o cluster do MSK na região primária com a posição de início do replicador definida para mais antigo. Isso é necessário para copiar os dados que você gravará na região secundária de volta para a região primária, para que você possa fazer failback para a região primária após o término do evento de serviço. Se você não definir a posição de início do replicador como o mais antigo, todos os dados produzidos para o cluster na região secundária durante o evento de serviço na região primária não serão copiados de volta para o cluster na região primária.

Realizar o failback para a região primária da AWS

Você pode retornar à AWS região primária após o término do evento de serviço nessa região.

Se você estiver usando a configuração de replicação de nomes de tópicos idênticos, siga estas etapas:

1. Crie um Replicador do MSK com o cluster secundário como origem e o cluster primário como destino e a posição de início definida para a replicação mais antiga de nomes de tópicos idênticos (mantenha o mesmo nome de tópicos no console).

   Isso iniciará o processo de cópia de todos os dados gravados no cluster secundário após o failover de volta para a região primária.
2. Monitore a `MessageLag` métrica no novo replicador na Amazon CloudWatch até que ela chegue`0`, o que indica que todos os dados foram replicados do secundário para o primário.
3. Depois que todos os dados tiverem sido replicados, interrompa a conexão de todos os produtores com o cluster secundário e inicie a conexão dos produtores com o cluster primário.
4. Aguarde a métrica `MaxOffsetLag` de seus consumidores que se conectam ao cluster secundário `0` para garantir que eles tenham processado todos os dados. Consulte Monitorar atrasos do consumidor.
5. Depois que todos os dados forem processados, interrompa os consumidores na região secundária e inicie a conexão dos consumidores ao cluster primário para concluir o failback.
6. Exclua o replicador que você criou na primeira etapa que está replicando dados do seu cluster secundário para o primário.
7. Verifique se o replicador existente que copia dados do cluster primário para o secundário tem o status “RUNNING” e a `ReplicatorThroughput` métrica na Amazon CloudWatch `0`.

   Observe que quando você cria um novo replicador com a posição de início como Mais antigo para failback, ele começa a ler todos os dados nos tópicos dos clusters secundários. Dependendo das configurações de retenção de dados, os tópicos podem ter dados provenientes do cluster de origem. Embora o Replicador do MSK filtre automaticamente essas mensagens, você ainda incorrerá em cobranças de processamento e transferência de dados para todos os dados no cluster secundário. Você pode rastrear o total de dados processados pelo replicador usando `ReplicatorBytesInPerSec`. Consulte Métricas de replicador do MSK.

Se você estiver usando a configuração de nomes de tópicos prefixados, siga estas etapas:

Você deve iniciar as etapas de failback somente depois que a replicação do cluster na região secundária para o cluster na região primária for recuperada e a métrica MessageLag na Amazon CloudWatch estiver próxima de 0. Um failback planejado não deve resultar em nenhuma perda de dados.

1. Feche todos os produtores e consumidores que se conectam ao cluster do MSK na região secundária.
2. Para a topologia ativa-passiva, exclua o replicador que está replicando dados do cluster na região secundária para a região primária. Você não precisa excluir o replicador para a topologia ativa-ativa.
3. Inicie a conexão dos produtores com o cluster do MSK na região primária.
4. Dependendo dos requisitos de ordenação de mensagens da aplicação, siga as etapas em uma das guias a seguir.

   No message ordering

   Se seu aplicativo não exigir a ordenação de mensagens, inicie consumidores na AWS região primária que leiam os tópicos locais (por exemplo,`topic`) e replicados (por exemplo,`<sourceKafkaClusterAlias>.topic`) usando um operador curinga (por exemplo,). `.*topic` Os consumidores de tópicos locais (p. ex., tópico) retomarão com base no último deslocamento que consumiram antes do failover. Se houver algum dado não processado antes do failover, ele será processado agora. No caso de um failover planejado, esse registro não deverá existir.

   Message ordering

   1. Inicie os consumidores somente para os tópicos replicados na região primária (p. ex., `<sourceKafkaClusterAlias>.topic`), mas não para os tópicos locais (p. ex., `topic`).
   2. Aguarde até que todos os consumidores de tópicos replicados na região primária do cluster concluam o processamento de todos os dados, para que o atraso do deslocamento seja 0 e o número de registros processados também seja 0. Em seguida, interrompa os consumidores dos tópicos replicados no cluster na região primária. Nesse ponto, todos os registros que foram produzidos na região secundária após o failover terão sido consumidos na região primária.
   3. Inicie consumidores para os tópicos locais (p. ex., `topic`) no cluster na região primária.

5. Verifique se o replicador existente do cluster na região primária para o cluster na região secundária está no estado EM EXECUÇÃO e funcionando conforme o esperado usando as métricas de latência e `ReplicatorThroughput`.

Criar uma configuração ativa-ativa usando o Replicador do MSK

Caso queira criar uma configuração ativa-ativa em que os dois clusters do MSK estejam servindo ativamente leituras e gravações, recomendamos que você use um Replicador do MSK com replicação de nomes de tópicos prefixados (Adicionar prefixo ao nome de tópicos no console). No entanto, isso exigirá que você reconfigure os consumidores para ler os tópicos replicados.

Siga estas etapas para configurar a topologia ativa-ativa entre o cluster A de origem do MSK e o cluster B de destino do MSK.

1. Crie um replicador do MSK com o cluster A do MSK como origem e o cluster B do MSK como destino.
2. Depois que o replicador do MSK acima for criado com sucesso, crie um replicador com o cluster B como origem e o cluster A como destino.
3. Crie dois conjuntos de produtores, cada um gravando dados ao mesmo tempo no tópico local (p. ex., “topic”) no cluster na mesma região do produtor.
4. Crie dois conjuntos de consumidores, cada um lendo dados usando uma assinatura curinga (como”. *tópico”) do cluster MSK na mesma AWS região do consumidor. Dessa forma, seus consumidores lerão automaticamente os dados produzidos localmente na região com base no tópico local (p. ex., `topic`), bem como os dados replicados de outra região no tópico com o prefixo`<sourceKafkaClusterAlias>.topic`. Esses dois conjuntos de consumidores devem ter grupos de consumidores diferentes IDs para que as compensações do grupo de consumidores não sejam sobrescritas quando o MSK Replicator as copia para o outro cluster.

Se quiser evitar a reconfiguração dos clientes, em vez da replicação do nome de tópicos prefixados (Adicionar prefixo ao nome de tópicos no console), você pode criar os Replicadores do MSK usando a replicação de nomes de tópicos (mantenha o mesmo nome de tópicos no console) para criar uma configuração ativa-ativa. No entanto, você pagará taxas adicionais de processamento e transferência de dados para cada replicador. Isso ocorre porque cada replicador precisará processar o dobro da quantidade normal de dados: uma vez para a replicação e outra para evitar loops infinitos. Você pode rastrear a quantidade total de dados processados por cada replicador usando a métrica `ReplicatorBytesInPerSec`. Consulte Monitorar a replicação. Essa métrica inclui os dados

replicados para o cluster de destino, bem como os dados filtrados pelo Replicador do MSK para evitar que os dados sejam copiados de volta para o mesmo tópico de origem.

> **Note**
>
> Se você estiver usando a replicação de nomes de tópicos idênticos (mantenha o mesmo nome de tópicos no console) para configurar a topologia ativa-ativa, aguarde pelo menos 30 segundos após excluir um tópico antes de recriar outro tópico com o mesmo nome. Esse período de espera ajuda a evitar que mensagens duplicadas sejam replicadas de volta para o cluster de origem. Seus consumidores devem ser capazes de reprocessar mensagens duplicadas sem impacto posterior. Consulte Considerações para criar aplicações do Apache Kafka em várias regiões.

Migrar um cluster do Amazon MSK para outro usando o Replicador do MSK

Você pode usar a replicação de nomes de tópicos idênticos para a migração de clusters, mas os consumidores devem ser capazes de lidar com mensagens duplicadas sem um impacto downstream. Isso ocorre porque o MSK Replicator fornece at-least-once replicação, o que pode levar à duplicação de mensagens em cenários raros. Se os consumidores atenderem a esse requisito, siga estas etapas.

1. Crie um replicador que replique dados do cluster antigo para o novo cluster com a posição de início do replicador definida como Mais antigo e usando a replicação de nomes de tópicos idênticos (mantenha o mesmo nome de tópicos no console).
2. Defina as configurações e permissões no nível de cluster no novo cluster. Você não precisa definir configurações em nível de tópico e leitura “literal”, pois o MSK Replicator ACLs as copia automaticamente.
3. Monitore a `MessageLag` métrica na Amazon CloudWatch até chegar a 0, o que indica que todos os dados foram replicados.
4. Depois que todos os dados tiverem sido replicados, impeça que os produtores gravem dados no cluster antigo.
5. Reconfigure esses produtores para se conectarem ao novo cluster e iniciá-los.
6. Monitore a métrica `MaxOffsetLag` dos consumidores lendo dados do cluster antigo até que eles cheguem a `0`, o que indica que todos os dados existentes foram processados.

7. Interrompa os consumidores que estão se conectando ao cluster antigo.
8. Reconfigure os consumidores para se conectarem ao novo cluster e iniciá-los.

Migre do autogerenciado MirrorMaker 2 para o MSK Replicator

Para migrar de MirrorMaker (MM2) para o MSK Replicator, siga estas etapas:

1. Interrompa o produtor que está gravando no cluster do Amazon MSK de origem.
2. Permita MM2 replicar todas as mensagens nos tópicos dos seus clusters de origem. Você pode monitorar o atraso do consumidor em relação MM2 ao consumidor em seu cluster MSK de origem para determinar quando todos os dados foram replicados.
3. Crie um novo replicador com a posição de início definida como Mais recente e a configuração dos nomes de tópicos definida como `IDENTICAL` (replicação dos mesmos nomes de tópicos no console).
4. Quando o replicador estiver no estado EM EXECUÇÃO, você poderá reiniciar a gravação dos produtores no cluster de origem.

Solucionar problemas do Replicador do MSK

As informações a seguir podem ajudar você a solucionar problemas que você pode vir a enfrentar com o replicador do MSK. Consulte Solução de problemas para o cluster do Amazon MSK para obter informações sobre a solução de problemas para outros recursos do Amazon MSK. Você também pode publicar seu problema no AWS re:Post.

O estado do replicador do MSK vai de CREATING para FAILED

Aqui estão algumas causas comuns de falha na criação do replicador do MSK.

1. Verifique se os grupos de segurança que você forneceu para a criação do replicador na seção do cluster de destino têm regras de saída para permitir o tráfego para os grupos de segurança do seu cluster de destino. Além disso, verifique se os grupos de segurança do seu cluster de destino têm regras de entrada que aceitem o tráfego proveniente dos grupos de segurança fornecidos para a criação do replicador na seção do cluster de destino. Consulte Escolher seu cluster de destino.
2. Se você estiver criando o replicador para replicação entre regiões, verifique se o cluster de origem tem conectividade multi-VPC ativada para o método de autenticação IAM Access Control.

Consulte Conectividade privada multi-VPC do Amazon MSK em uma única região. Verifique também se a política de cluster está configurada no cluster de origem para que o replicador do MSK possa se conectar ao cluster de origem. Consulte Preparar o cluster de origem do Amazon MSK.

3. Verifique se a perfil do IAM que você forneceu durante a criação do replicador do MSK tem as permissões necessárias para ler e gravar nos clusters de origem e destino. Além disso, verifique se a perfil do IAM tem permissões para gravar em tópicos. Consulte Definir configurações e permissões do replicador
4. Verifique se sua rede não ACLs está bloqueando a conexão entre o MSK Replicator e seus clusters de origem e destino.
5. É possível que os clusters de origem ou de destino não estivessem totalmente disponíveis quando o replicador do MSK tentou se conectar a eles. Isso pode decorrer de níveis excessivos de carga, uso do disco ou da CPU, o que faz com que o replicador não consiga se conectar aos agentes. Corrija o problema com os agentes e repita a criação do replicador.

Após realizar as validações acima, crie o replicador do MSK novamente.

O replicador do MSK parece preso no estado CREATING

Às vezes a criação do replicador do MSK pode levar até 30 minutos. Aguarde 30 minutos e verifique o estado do replicador novamente.

O replicador do MSK não está replicando dados ou replicando apenas dados parciais

Siga estas etapas para solucionar problemas de replicação de dados.

1. Verifique se seu replicador não está enfrentando nenhum erro de autenticação usando a AuthError métrica fornecida pelo MSK Replicator na Amazon. CloudWatch Se essa métrica estiver acima de 0, verifique se a política do perfil do IAM que você forneceu para o replicador é válida e se não há recusa de permissões definidas para as permissões do cluster. Com base na dimensão ClusterAlias, você pode identificar se o cluster de origem ou de destino está apresentando erros de autenticação.
2. Verifique se seus clusters de origem e destino não estão enfrentando problemas. É possível que o replicador não consiga se conectar ao seu cluster de origem ou de destino. Isso pode acontecer devido a muitas conexões, disco com capacidade total ou alto uso da CPU.

3. Verifique se seus clusters de origem e destino podem ser acessados pelo MSK Replicator usando a métrica KafkaClusterPingSuccessCount na Amazon. CloudWatch Com base na dimensão ClusterAlias, você pode identificar se o cluster de origem ou de destino está apresentando erros de autenticação. Se essa métrica for 0 ou não tiver ponto de dados, a conexão não está íntegra. Você deve verificar as permissões de rede e do perfil do IAM que o replicador do MSK está usando para se conectar aos seus clusters.
4. Verifique se seu replicador não está enfrentando falhas devido à falta de permissões em nível de tópico usando a métrica ReplicatorFailure na Amazon. CloudWatch Se essa métrica estiver acima de 0, verifique o perfil do IAM que você forneceu para obter permissões no nível de tópico.
5. Verifique se a expressão regular que você forneceu na lista de permissões ao criar o replicador corresponde aos nomes dos tópicos que você deseja replicar. Além disso, verifique se os tópicos não estão sendo excluídos da replicação devido a uma expressão regular na lista de proibição.
6. Observe que pode levar até 30 segundos para que o replicador detecte e crie os novos tópicos ou partições de tópicos no cluster de destino. Qualquer mensagem produzida no tópico de origem antes da criação do tópico no cluster de destino não será replicada se a posição de início do replicador for a mais recente (padrão). Como alternativa, você poderá iniciar a replicação a partir do primeiro deslocamento nas partições de tópicos do cluster de origem se quiser replicar as mensagens existentes nos tópicos no cluster de destino. Consulte Definir configurações e permissões do replicador.

Deslocamentos de mensagens no cluster de destino são diferentes do cluster de origem

Como parte da replicação de dados, o Replicador do MSK consome mensagens do cluster de origem e as produz para o cluster de destino. Isso pode levar as mensagens a terem deslocamentos diferentes nos clusters de origem e de destino. No entanto, se você ativou a sincronização de deslocamentos de grupos de consumidores durante a criação do replicador, o Replicador do MSK converterá automaticamente os deslocamentos enquanto copia os metadados para que, após o failover para o cluster de destino, os consumidores possam retomar o processamento próximo de onde pararam no cluster de origem.

O Replicador do MSK não está sincronizando deslocamentos de grupos de consumidores ou o grupo de consumidores não existe no cluster de destino

Siga estas etapas para solucionar problemas de replicação de metadados.

1. Verifique se a replicação de dados está funcionando conforme esperado. Se não, consulte O replicador do MSK não está replicando dados ou replicando apenas dados parciais.
2. Verifique se a expressão regular que você forneceu na lista de permissões ao criar o replicador corresponde aos nomes dos grupos de consumidores que você deseja replicar. Além disso, verifique se os grupos de consumidores não estão sendo excluídos da replicação devido a uma expressão regular na lista de proibições.
3. Verifique se o Replicador do MSK criou o tópico no cluster de destino. Pode levar até 30 segundos para que o replicador detecte e crie os novos tópicos ou partições de tópicos no cluster de destino. Qualquer mensagem produzida no tópico de origem antes da criação do tópico no cluster de destino não será replicada se a posição de início do replicador for a mais recente (padrão). Se o grupo de consumidores no cluster de origem tiver consumido somente as mensagens que não foram replicadas pelo Replicador do MSK, o grupo de consumidores não será replicado para o cluster de destino. Depois que o tópico for criado com sucesso no cluster de destino, o Replicador do MSK começará a replicar mensagens recém-gravadas no cluster de origem para o destino. Quando o grupo de consumidores começar a ler essas mensagens da origem, o Replicador do MSK replicará automaticamente o grupo de consumidores para o cluster de destino. Como alternativa, você poderá iniciar a replicação a partir do primeiro deslocamento nas partições de tópicos do cluster de origem se quiser replicar as mensagens existentes nos tópicos no cluster de destino. Consulte Definir configurações e permissões do replicador.

> **Note**
>
> O Replicador do MSK otimiza a sincronização do deslocamento de grupos de consumidores para os consumidores no cluster de origem, que estão lendo de uma posição mais próxima ao final da partição do tópico. Se os grupos de consumidores estiverem em atraso no cluster de origem, você poderá observar um atraso maior para esses grupos de consumidores no destino em comparação com a origem. Isso significa que, após o failover para o cluster de destino, os consumidores reprocessarão mais mensagens duplicadas. Para reduzir esse atraso, os consumidores no cluster de origem precisariam se atualizar e começar a consumir a partir da ponta do stream (final da partição do tópico). À medida que os consumidores se atualizarem, o Replicador do MSK reduzirá automaticamente o atraso.

A latência de replicação é alta ou continua aumentando

Aqui estão algumas causas comuns da alta latência de replicação.

1. Verifique se você tem o número certo de partições nos clusters de origem e destino do MSK. Ter poucas ou muitas partições pode afetar o desempenho. Para obter orientação sobre como escolher o número de partições, consulte Práticas recomendadas para usar o replicador do MSK. A tabela a seguir mostra o número mínimo recomendado de partições para obter o throughput desejado com o replicador do MSK.

   Throughput e número mínimo recomendado de partições

Throughput (MB/s)	Número mínimo necessário de partições
50	167
100	334
250	833
500	1666
1000	3333

2. Verifique se você tem capacidade suficiente de leitura e gravação em seus clusters de origem e destino do MSK para atender o tráfego de replicação. O replicador do MSK atua como consumidor do cluster de origem (saída) e como produtor do cluster de destino (entrada). Portanto, você deve provisionar a capacidade do cluster para atender ao tráfego de replicação, além de outros tráfegos em seus clusters. Consulte ??? para obter orientação sobre como dimensionar seus clusters do MSK.
3. A latência de replicação pode variar para clusters MSK em diferentes pares de AWS regiões de origem e destino, dependendo da distância geográfica entre os clusters. Por exemplo, a latência de replicação geralmente é menor ao replicar entre clusters nas regiões da Europa (Irlanda) e Europa (Londres) em comparação com a replicação entre clusters nas regiões da Europa (Irlanda) e Ásia-Pacífico (Sydney).
4. Verifique se o replicador não está sendo submetido ao controle de utilização devido às cotas excessivamente agressivas definidas em seus clusters de origem ou de destino. Você pode usar a ThrottleTime métrica fornecida pelo MSK Replicator na Amazon CloudWatch para ver o tempo médio em milissegundos em que uma solicitação foi limitada pelos corretores em seu cluster. source/target Se essa métrica estiver acima de 0, você deve ajustar as cotas do Kafka para reduzir o controle de utilização de modo que o replicador possa se atualizar. Consulte Como gerenciar o throughput do replicador do MSK usando cotas do Kafka para obter informações sobre o gerenciamento de cotas do Kafka para o replicador.

5. ReplicationLatency e MessageLag pode aumentar quando uma AWS região se degrada. Use o AWS Service Health Dashboard para verificar se há um evento de serviço do MSK na região do seu cluster primário do MSK. Se houver um evento de serviço, você poderá redirecionar temporariamente as leituras e gravações da aplicação para a outra região.

Solução de problemas de falhas do MSK Replicator usando métricas ReplicatorFailure

A ReplicatorFailure métrica ajuda você a monitorar e detectar problemas de replicação no MSK Replicator. Um valor diferente de zero dessa métrica normalmente indica um problema de falha na replicação, que pode resultar dos seguintes fatores:

- limitações de tamanho da mensagem
- violações do intervalo de timestamp
- registrar problemas de tamanho de lote

Se a ReplicatorFailure métrica relatar um valor diferente de zero, siga estas etapas para solucionar o problema.

> **Note**
>
> Para obter mais informações sobre essa métrica, consulte Métricas de replicador do MSK.

1. Configure um cliente que seja capaz de se conectar ao cluster MSK de destino e tenha as ferramentas de CLI do Apache Kafka configuradas. Para obter informações sobre como configurar o cliente e a ferramenta Kafka CLI, consulte. Conecte-se a um cluster provisionado do Amazon MSK
2. Abra o console Amazon MSK em https://console.aws.amazon.com/msk/casa? region=us-east-1#/home/.

   Faça o seguinte:

   a. Obtenha o ARNs MSK Replicator e o cluster MSK de destino.

   b. Obtenha os endpoints do broker do cluster MSK de destino. Você usará esses endpoints nas etapas a seguir.

3. Execute os comandos a seguir para exportar o ARN do MSK Replicator e os endpoints do broker que você obteve na etapa anterior.

   Certifique-se de substituir os valores de espaço reservado para < *ReplicatorARN* >, < *BootstrapServerString* > e < *ConsumerConfigFile* > usados nos exemplos a seguir por seus valores reais.

   ```
   export TARGET_CLUSTER_SERVER_STRING=<BootstrapServerString>
   ```

   ```
   export REPLICATOR_ARN=<ReplicatorARN>
   ```

   ```
   export CONSUMER_CONFIG_FILE=<ConsumerConfigFile>
   ```

4. No seu <*path-to-your-kafka-installation*>/bin diretório, faça o seguinte:

   a. Salve o script a seguir e dê um nome a ele**query-replicator-failure-message.sh**.

   ```
   #!/bin/bash

   # Script: Query MSK Replicator Failure Message
   # Description: This script queries exceptions from AWS MSK Replicator status
    topics
   # It takes a replicator ARN and bootstrap server as input and searches for
    replicator exceptions
   # in the replicator's status topic, formatting and displaying them in a
    readable manner
   #
   # Required Arguments:
   #   --replicator-arn: The ARN of the AWS MSK Replicator
   #   --bootstrap-server: The Kafka bootstrap server to connect to
   #   --consumer.config: Consumer config properties file
   # Usage Example:
   #   ./query-replicator-failure-message.sh ./query-replicator-failure-message.sh
    --replicator-arn <replicator-arn> --bootstrap-server <bootstrap-server> --
   consumer.config <consumer.config>

   print_usage() {
     echo "USAGE: $0 ./query-replicator-failure-message.sh --replicator-arn
    <replicator-arn> --bootstrap-server <bootstrap-server> --consumer.config
    <consumer.config>"
   ```

```
  echo "--replicator-arn <String: MSK Replicator ARN>      REQUIRED: The ARN of
 AWS MSK Replicator."
  echo "--bootstrap-server <String: server to connect to>  REQUIRED: The Kafka
 server to connect to."
  echo "--consumer.config <String: config file>           REQUIRED: Consumer
 config properties file."
  exit 1
}

# Initialize variables
replicator_arn=""
bootstrap_server=""
consumer_config=""

# Parse arguments
while [[ $# -gt 0 ]]; do
  case "$1" in
    --replicator-arn)
      if [ -z "$2" ]; then
        echo "Error: --replicator-arn requires an argument."
        print_usage
      fi
      replicator_arn="$2"; shift 2 ;;
    --bootstrap-server)
      if [ -z "$2" ]; then
        echo "Error: --bootstrap-server requires an argument."
        print_usage
      fi
      bootstrap_server="$2"; shift 2 ;;
    --consumer.config)
      if [ -z "$2" ]; then
        echo "Error: --consumer.config requires an argument."
        print_usage
      fi
      consumer_config="$2"; shift 2 ;;
    *) echo "Unknown option: $1"; print_usage ;;
  esac
done

# Check for required arguments
if [ -z "$replicator_arn" ] || [ -z "$bootstrap_server" ] || [ -z
 "$consumer_config" ]; then
  echo "Error: --replicator-arn, --bootstrap-server, and --consumer.config are
 required."
```

```
  print_usage
fi

# Extract replicator name and suffix from ARN
replicator_arn_suffix=$(echo "$replicator_arn" | awk -F'/' '{print $NF}')
replicator_name=$(echo "$replicator_arn" | awk -F'/' '{print $(NF-1)}')
echo "Replicator name: $replicator_name"

# List topics and find the status topic
topics=$(./kafka-topics.sh --command-config client.properties --list --
bootstrap-server "$bootstrap_server")
status_topic_name="__amazon_msk_replicator_status_${replicator_name}_
${replicator_arn_suffix}"

# Check if the status topic exists
if echo "$topics" | grep -Fq "$status_topic_name"; then
  echo "Found replicator status topic: '$status_topic_name'"
  ./kafka-console-consumer.sh --bootstrap-server "$bootstrap_server" --
consumer.config "$consumer_config" --topic "$status_topic_name" --from-
beginning | stdbuf -oL grep "Exception" | stdbuf -oL sed -n 's/.*Exception:\(.*
\) Topic: \([^,]*\), Partition: \([^\]*\).*/ReplicatorException:\1 Topic: \2,
 Partition: \3/p'
else
  echo "No topic matching the pattern '$status_topic_name' found."
fi
```

b. Execute esse script para consultar as mensagens de falha do 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
```

Esse script gera todos os erros com suas mensagens de exceção e partições de tópicos afetadas. Você pode usar essas informações de exceção para mitigar as falhas, conforme descrito emFalhas comuns do MSK Replicator e suas soluções. Como o tópico contém todas as mensagens históricas de falha, inicie a investigação usando a última mensagem. Veja a seguir um exemplo de uma mensagem de falha.

```
ReplicatorException: The request included a message larger than the max message
 size the server will accept. Topic: test, Partition: 1
```

Falhas comuns do MSK Replicator e suas soluções

A lista a seguir descreve algumas das falhas do MSK Replicator que você pode enfrentar e como mitigá-las.

Tamanho da mensagem maior que max.request.size

Causa

Essa falha ocorre quando o MSK Replicator não consegue replicar os dados porque o tamanho da mensagem individual excede 10 MB. Por padrão, o MSK Replicator replica mensagens de até 10 MB de tamanho.

Veja a seguir um exemplo desse tipo de mensagem de falha.

```
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
```

Solução

Reduza o tamanho das mensagens individuais em seu tópico. Se você não conseguir fazer isso, siga estas instruções para solicitar um aumento de limite.

Tamanho da mensagem maior que o tamanho máximo da mensagem que o servidor aceitará

Causa

Essa falha ocorre quando o tamanho da mensagem excede o tamanho máximo da mensagem do cluster de destino.

Veja a seguir um exemplo desse tipo de mensagem de falha.

```
ReplicatorException: The request included a message larger than the max message size
 the server will accept. Topic: test, Partition: 1
```

Solução

Aumente a `max.message.bytes` configuração no cluster de destino ou no tópico correspondente do cluster de destino. Defina a `max.message.bytes` configuração do cluster

de destino para corresponder ao seu maior tamanho de mensagem não compactada. Para obter informações sobre como fazer isso, consulte max.message.bytes.

O carimbo de data/hora está fora do alcance

Causa

Essa falha ocorre porque o timestamp da mensagem individual está fora do intervalo permitido do cluster de destino.

Veja a seguir um exemplo desse tipo de mensagem de falha.

```
ReplicatorException: Timestamp 1730137653724 of message with offset 0 is out of
 range. The timestamp should be within [1730137892239, 1731347492239] Topic: test,
 Partition: 1
```

Solução

Atualize a `message.timestamp.before.max.ms` configuração do cluster de destino para permitir mensagens com carimbos de data/hora mais antigos. Para obter informações sobre como fazer isso, consulte message.timestamp.before.max.ms.

Grave um lote muito grande

Causa

Essa falha ocorre porque o tamanho do lote de registros excede o tamanho do segmento definido para o tópico no cluster de destino. O MSK Replicator suporta um tamanho máximo de lote de 1 MB.

Veja a seguir um exemplo desse tipo de mensagem de falha.

```
ReplicatorException: The request included message batch larger than the configured
 segment size on the server. Topic: test, Partition: 1
```

Solução

A configuração segment.bytes do cluster de destino deve ser pelo menos tão grande quanto o tamanho do lote (1 MB) para que o Replicator continue sem erros. Atualize o segment.bytes do cluster de destino para ter pelo menos 1048576 (1 MB). Para obter informações sobre como fazer isso, consulte segment.bytes.

> **Note**
>
> Se a ReplicatorFailure métrica continuar emitindo valores diferentes de zero após a aplicação dessas soluções, repita o processo de solução de problemas até que a métrica emita um valor zero.

Práticas recomendadas para usar o replicador do MSK

Esta seção aborda práticas recomendadas e estratégias de implementação comuns para usar o Replicador do Amazon MSK.

Tópicos

Como gerenciar o throughput do replicador do MSK usando cotas do Kafka

Como o replicador do MSK atua como consumidor do seu cluster de origem, a replicação pode fazer com que outros consumidores passem por controle de utilização em seu cluster de origem. A quantidade de controle de utilização depende da capacidade de leitura que você tem no cluster de origem e do throughput dos dados que você está replicando. Recomendamos que você provisione capacidade idêntica para seus clusters de origem e de destino e leve em conta o throughput de replicação ao calcular a capacidade necessária.

Você também pode definir cotas do Kafka para o replicador em seus clusters de origem e destino a fim de controlar a capacidade que o replicador do MSK pode usar. Recomenda-se usar uma cota de largura de banda da rede. Uma cota de largura de banda da rede define um limite de taxa de bytes, definido como bytes por segundo, para um ou mais clientes que compartilham uma cota. Essa cota é definida por agente.

Siga estas etapas para aplicar uma cota.

1. Recupere a string do servidor bootstrap para o cluster de origem. Consulte Obter os agentes de bootstrap para um cluster do Amazon MSK.

2. Recupere o Service execution role (SER – Perfil de execução de serviço) usado pelo replicador do MSK. Esse é o SER que você usou para uma solicitação `CreateReplicator`. Você também pode extrair o SER da DescribeReplicator resposta de um replicador existente.
3. Usando as ferramentas de CLI do Kafka, execute o comando a seguir no cluster de origem.

```
./kafka-configs.sh --bootstrap-server <source-cluster-bootstrap-server> --alter --
add-config 'consumer_byte_
rate=<quota_in_bytes_per_second>' --entity-type users --entity-name
 arn:aws:sts::<customer-account-id>:assumed-role/<ser-role-name>/<customer-account-
id> --command-config <client-properties-for-iam-auth></programlisting>
```

4. Após executar o comando acima, verifique se a métrica `ReplicatorThroughput` não ultrapassa a cota que você definiu.

Observe que todos estarão sujeitos a essa cota se você reutilizar um perfil de execução de serviço entre vários replicadores do MSK. Se você quiser manter cotas separadas por replicador, use perfis de execução de serviço separados.

Para obter mais informações sobre o uso da autenticação do IAM no MSK com cotas, consulte Clusters Apache Kafka multilocação no Amazon MSK com controle de acesso do IAM e cotas do Kafka: parte 1.

> **Warning**
>
> Definir uma taxa de consumer_byte_rate extremamente baixa pode fazer com que seu replicador do MSK atue de maneiras inesperadas.

Definir o período de retenção do cluster

Você pode definir o período de retenção de log para clusters do MSK provisionados e com tecnologia sem servidor. O período recomendado de retenção é de 7 dias. Consulte Alterações na configuração de cluster ou Configuração do cluster compatível do MSK Sem Servidor.

Integrações do MSK

Esta seção fornece referências aos AWS recursos que se integram ao Amazon MSK.

Tópicos

Conector do Amazon Athena para o Amazon MSK

O conector do Amazon Athena para o Amazon MSK possibilita que o Amazon Athena execute consultas SQL em tópicos do Apache Kafka. Use esse conector para visualizar os tópicos e as mensagens do Apache Kafka no Athena como tabelas e linhas, respectivamente.

Para obter mais informações, consulte Conector do MSK para Amazon Athena no Guia do usuário do Amazon Athena.

Ingestão de dados de streaming do Amazon Redshift para Amazon MSK

A ingestão de streaming do Amazon Redshift oferece suporte ao Amazon MSK. O recurso de ingestão de streaming do Amazon Redshift fornece ingestão de dados com baixa latência e alta velocidade do Amazon MSK para uma visão materializada do Amazon Redshift. Como não precisa armazenar dados no Amazon S3, o Amazon Redshift pode ingerir dados de streaming com uma latência menor e com um custo de armazenamento reduzido. Você pode configurar a ingestão de streaming do Amazon Redshift em um cluster do Amazon Redshift usando instruções SQL para autenticar e se conectar a um tópico do Amazon MSK.

Para obter mais informações, consulte Ingestão de streaming no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Integração do Firehose para Amazon MSK

O Amazon MSK se integra ao Firehose para fornecer uma solução sem servidor e sem código para entregar streams dos clusters do Apache Kafka para data lakes do Amazon S3. O Firehose é um serviço de extração, transformação e carregamento (ETL) de streaming que lê dados dos tópicos do Amazon MSK Kafka, realiza transformações, como conversão para Parquet, e agrega e grava os dados no Amazon S3. Com alguns cliques no console, você pode configurar um stream do Firehose para ler um tópico do Kafka e entregá-lo em um local do S3. Não há código para escrever, aplicações de conectores nem recursos para provisionar. O Firehose escala automaticamente com base na quantidade de dados publicados no tópico do Kafka, e você paga somente pelos bytes ingeridos do Kafka.

Veja mais informações sobre esse recurso nos itens abaixo.

- Writing to Kinesis Data Firehose Using Amazon MSK - Amazon Kinesis Data Firehose no Guia do desenvolvedor do Amazon Kinesis Data Firehose
- Blog: Amazon MSK apresenta entrega gerenciada de dados do Apache Kafka para seu data lake
- Laboratório: Delivery to Amazon S3 using Firehose

Acesse o Amazon EventBridge Pipes por meio do console Amazon MSK

O Amazon EventBridge Pipes conecta as fontes aos alvos. Os tubos são destinados a point-to-point integrações entre fontes e alvos suportados, com suporte para transformações e enriquecimento avançados. EventBridge O Pipes fornece uma maneira altamente escalável de conectar seu cluster Amazon MSK a AWS serviços como Step Functions, Amazon SQS e API Gateway, bem como a aplicativos de software como serviço (SaaS) de terceiros, como o Salesforce.

Para configurar um pipe, você escolhe a origem, adiciona filtragem opcional, define o enriquecimento opcional e escolhe o destino para os dados do evento.

Na página de detalhes do cluster do Amazon MSK, você pode ver os pipes que usam esse cluster como origem. Nessa página, você também pode:

- Inicie o EventBridge console para ver os detalhes do tubo.
- Inicie o EventBridge console para criar um novo canal com o cluster como fonte.

Para obter mais informações sobre como configurar um cluster Amazon MSK como fonte de canal, consulte o cluster Amazon Managed Streaming for Apache Kafka como fonte no Guia do usuário da Amazon. EventBridge Para obter mais informações sobre EventBridge tubos em geral, consulte EventBridge Tubos.

Para acessar EventBridge canais para um determinado cluster Amazon MSK

1. Abra o Console do Amazon MSK e selecione Clusters.
2. Selecione um cluster.
3. Na página de detalhes do cluster, escolha a guia Integração.

   A guia Integração inclui uma lista de todos os pipes configurados para usar o cluster selecionado como origem, inclusive:

   - nome do pipe
   - status atual
   - destino do pipe
   - última modificação do pipe
4. Gerencie os pipes do seu cluster do Amazon MSK conforme desejado:

   Para acessar mais detalhes sobre um pipe

   - Escolha o pipe.

     Isso abre a página de detalhes do Pipe do EventBridge console.

   Para criar um pipe

   - Escolha Conectar cluster do Amazon MSK ao Pipe.

     Isso inicia a página Create pipe do EventBridge console, com o cluster Amazon MSK especificado como a origem do pipe. Para obter mais informações, consulte EventBridge Criação de um tubo no Guia EventBridge do usuário da Amazon.

   - Você também pode criar um canal para um cluster na página Clusters. Selecione o cluster e, no menu Ações, selecione Create EventBridge Pipe.

Usando o Kafka Streams com corretores MSK Express e MSK Serverless

O Kafka Streams suporta transformações sem estado e com estado. Transformações com estado, como contar, agregar ou unir, usam operadores que armazenam seu estado em tópicos internos do Kafka. Além disso, algumas transformações sem estado, como groupBy ou repartição, armazenam seus resultados em tópicos internos do Kafka. Por padrão, o Kafka Streams nomeia esses tópicos internos com base no operador correspondente. Se esses tópicos não existirem, o Kafka Streams cria tópicos internos do Kafka. Para criar os tópicos internos, o Kafka Streams codifica a configuração segment.bytes e a define para 50 MB. O MSK Provisioned with Express Brokers e o MSK Serverless protege algumas configurações de tópicos, incluindo segment.size durante a criação do tópico. Portanto, um aplicativo Kafka Streams com transformações com estado falha em criar os tópicos internos usando os corretores MSK Express ou o MSK Serverless.

Para executar esses aplicativos do Kafka Streams em corretores MSK Express ou MSK Serverless, você mesmo deve criar os tópicos internos. Para fazer isso, primeiro identifique e nomeie os operadores do Kafka Streams, que exigem tópicos. Em seguida, crie os tópicos internos correspondentes do Kafka.

> **Note**
>
> - É uma boa prática nomear os operadores manualmente no Kafka Streams, especialmente aqueles que dependem de tópicos internos. Para obter informações sobre como nomear operadores, consulte Nomeando operadores em um aplicativo DSL do Kafka Streams na documentação do Kafka Streams.
> - O nome do tópico interno para uma transformação com estado depende `application.id` do aplicativo Kafka Streams e do nome do operador com estado,. `application.id-statefuloperator_name`

Tópicos

- Criação de um aplicativo Kafka Streams usando corretores MSK Express ou MSK Serverless

Criação de um aplicativo Kafka Streams usando corretores MSK Express ou MSK Serverless

Se seu aplicativo Kafka Streams estiver `application.id` configurado como`msk-streams-processing`, você poderá criar um aplicativo Kafka Streams usando corretores MSK Express ou MSK Serverless. Para fazer isso, use o `count()` operador, que requer um tópico interno com o nome. Por exemplo, `msk-streams-processing-count-store`.

Para criar um aplicativo Kafka Streams, faça o seguinte:

Tópicos

- Identifique e nomeie os operadores
- Crie os tópicos internos
- (Opcional) Verifique o nome do tópico
- Exemplos de operadores de nomenclatura

Identifique e nomeie os operadores

1. Identifique os processadores com estado usando as transformações com estado na documentação do Kafka Streams.

   Alguns exemplos de processadores com estado incluem `count`, `aggregate`, ou`join`.

2. Identifique os processadores que criam tópicos para reparticionamento.

   O exemplo a seguir contém uma `count()` operação que precisa de um estado.

   ```
   var stream =
       paragraphStream
           .groupByKey()
           .count()
           .toStream();
   ```

3. Para nomear o tópico, adicione um nome para cada processador com estado. Com base no tipo de processador, a nomenclatura é feita por uma classe de nomenclatura diferente. Por exemplo, a `count()` operação é uma operação de agregação. Portanto, ele precisa da `Materialized` classe.

Para obter informações sobre as classes de nomenclatura para as operações com estado, consulte Conclusão na documentação do Kafka Streams.

O exemplo a seguir define o nome do `count()` operador para `count-store` usar a `Materialized` classe.

```
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();
```

Crie os tópicos internos

O Kafka Streams prefixos `application.id` para nomes de tópicos internos, onde é definido pelo usuário. `application.id` Por exemplo, `application.id-internal_topic_name`. Os tópicos internos são tópicos normais do Kafka, e você pode criar os tópicos usando as informações disponíveis na API `AdminClient` do Kafka Criar um tópico do Apache Kafka ou dela.

Dependendo do seu caso de uso, você pode usar as políticas padrão de limpeza e retenção do Kafka Streams ou personalizar seus valores. Você os define em `cleanup.policy` `retention.ms` e.

O exemplo a seguir cria os tópicos com a `AdminClient` API e define **`msk-streams-processing`** o. `application.id`

```
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);
}
```

Depois que os tópicos forem criados no cluster, seu aplicativo Kafka Streams poderá usar o `msk-streams-processing-count-store` tópico para a operação. `count()`

(Opcional) Verifique o nome do tópico

Você pode usar o descritor de topografia para descrever a topologia do seu stream e visualizar os nomes dos tópicos internos. O exemplo a seguir mostra como executar o descritor de topologia.

```
final StreamsBuilder builder = new StreamsBuilder();
Topology topology = builder.build();
System.out.println(topology.describe());
```

A saída a seguir mostra a topologia do fluxo para o exemplo 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 obter informações sobre como usar o descritor de topologia, consulte Nomeando operadores em um aplicativo DSL do Kafka Streams na documentação do Kafka Streams.

Exemplos de operadores de nomenclatura

Esta seção fornece alguns exemplos de operadores de nomenclatura.

Exemplo de operador de nomenclatura para groupByKey ()

```
groupByKey() -> groupByKey(Grouped.as("kafka-stream-groupby"))
```

Exemplo de operador de nomenclatura para contagem normal ()

```
normal count() -> .count(Materialized.<String, Long, KeyValueStore<Bytes,
 byte[]>>as("kafka-streams-window")  // descriptive name for the store
    .withKeySerde(Serdes.String())
```

```
    .withValueSerde(Serdes.Long()))
```

Exemplo de operador de nomenclatura para contagem em janela ()

```
windowed count() -> .count(Materialized.<String, Long, WindowStore<Bytes,
 byte[]>>as("kafka-streams-window")  // descriptive name for the store
    .withKeySerde(Serdes.String())
    .withValueSerde(Serdes.Long()))
```

Exemplo de operador de nomenclatura para windowed suppressed ()

```
windowed suppressed() ->
Suppressed<Windowed> suppressed = Suppressed
        .untilWindowCloses(Suppressed.BufferConfig.unbounded())
        .withName("kafka-suppressed");
    .suppress(suppressed)
```

Planos de incorporação de vetores em tempo real

O Amazon MSK (Managed Streaming for Apache Kafka) oferece suporte ao Amazon Managed Service para esquemas do Apache Flink para gerar incorporações vetoriais usando o Amazon Bedrock, simplificando o processo de criação de aplicativos de IA em tempo real baseados em dados contextuais. up-to-date O plano do MSF simplifica o processo de incorporação dos dados mais recentes de seus pipelines de streaming do Amazon MSK em seus modelos generativos de IA, eliminando a necessidade de escrever código personalizado para integrar fluxos de dados em tempo real, bancos de dados vetoriais e grandes modelos de linguagem.

Você pode configurar o esquema do MSF para gerar continuamente incorporações vetoriais usando os modelos de incorporação da Bedrock e, em seguida, indexar essas incorporações no Service OpenSearch para seus fluxos de dados do Amazon MSK. Isso permite combinar o contexto de dados em tempo real com os poderosos modelos de linguagem grande da Bedrock para gerar respostas precisas de up-to-date IA sem escrever código personalizado. Você também pode optar por melhorar a eficiência da recuperação de dados usando o suporte integrado para técnicas de agrupamento de dados da biblioteca de LangChain código aberto que oferece suporte a entradas de alta qualidade para ingestão de modelos. O blueprint gerencia a integração e o processamento de dados entre o MSK, o modelo de incorporação escolhido e o armazenamento OpenSearch vetorial, permitindo que você se concentre na criação de seus aplicativos de IA, em vez de gerenciar a integração subjacente.

Os blueprints de incorporação de vetores em tempo real estão disponíveis nas seguintes regiões da: AWS

- Norte da Virgínia: us-east-1
- Ohio, us-east-2
- Oregon, us-west-2
- Mumbai: ap-south-1
- Seul: ap-northeast-2
- Cingapura: ap-southeast-1 ap-southeast-1
- Sydney: ap-southeast-2
- Tóquio: ap-northeast-1
- Canadá Central: ca-central-1
- Frankfurt: eu-central-1
- Irlanda: eu-west-1
- Londres: eu-west-2
- Paris: eu-west-3
- São Paulo: sa-east-1

Tópicos

Registro e observabilidade

Todos os registros e métricas para esquemas de incorporação vetorial em tempo real podem ser ativados usando CloudWatch registros.

Todas as métricas que estão disponíveis para um aplicativo regular do MSF e do Amazon Bedrock podem monitorar seu aplicativo e as métricas do Bedrock.

Há duas métricas adicionais para monitorar o desempenho da geração de incorporações. Essas métricas fazem parte do nome da EmbeddingGeneration operação em CloudWatch.

- BedrockTitanEmbeddingTokenCount: monitora o número de tokens presentes em uma única solicitação ao Bedrock.
- BedrockEmbeddingGenerationLatencyMs: relata o tempo gasto para enviar e receber uma resposta da Bedrock para gerar incorporações em milissegundos.

Para OpenSearch Service, é possível usar as seguintes métricas:

- OpenSearch Métricas de coleta sem servidor: consulte Monitoramento OpenSearch sem servidor com a Amazon CloudWatch no Guia do desenvolvedor do OpenSearch Amazon Service.
- OpenSearch métricas provisionadas: consulte Monitoramento de métricas de OpenSearch cluster com a Amazon CloudWatch no Amazon OpenSearch Service Developer Guide.

Notas antes de ativar os esquemas de incorporação vetorial em tempo real

O aplicativo Managed Service for Apache Flink só suportará texto não estruturado ou dados JSON no fluxo de entrada.

Dois modos de processamento de entrada são suportados:

- Quando os dados de entrada são texto não estruturado, toda a mensagem de texto é incorporada. O banco de dados vetorial contém o texto original e a incorporação gerada.
- Quando os dados de entrada estão no formato JSON, o aplicativo permite que você configure e especifique uma ou mais chaves dentro do valor do objeto JSON para usar no processo de incorporação. Se houver mais de uma chave, todas as chaves serão vetorizadas juntas e indexadas no banco de dados vetorial. O banco de dados vetorial conterá a mensagem original e a incorporação gerada.

Geração de incorporação: O aplicativo suporta todos os modelos de incorporação de texto fornecidos exclusivamente pela Bedrock.

Persista no armazenamento de banco de dados vetorial: o aplicativo usa um OpenSearch cluster existente (provisionado ou sem servidor) na conta do cliente como destino para dados incorporados persistentes. Ao usar o Opensearch Serverless para criar um índice vetorial, sempre use o nome do campo vetorial. `embedded_data`

Semelhante aos blueprints do MSF, espera-se que você gerencie a infraestrutura para executar o código associado ao blueprint de incorporação vetorial em tempo real.

Semelhante ao MSF Blueprints, depois que um aplicativo MSF é criado, ele deve ser iniciado exclusivamente na AWS conta usando o console ou a CLI. AWS não iniciará o aplicativo MSF para você. Você precisa chamar a StartApplication API (por meio da CLI ou do console) para executar o aplicativo.

Movimentação de dados entre contas: o aplicativo não permite que você mova dados entre o fluxo de entrada e os destinos vetoriais que residem em AWS contas diferentes.

Implemente um plano de vetorização de dados de streaming

Este tópico descreve como implantar um blueprint de vetorização de dados de streaming.

Implemente um plano de vetorização de dados de streaming

1. Certifique-se de que os seguintes recursos estejam configurados corretamente:
   - Cluster MSK provisionado ou sem servidor com um ou mais tópicos contendo dados.
2. Configuração Bedrock: Acesso ao modelo Bedrock desejado. Os modelos Bedrock atualmente suportados são:
   - Amazon Titan Embeddings G1 - Text
   - Incorporador de Texto do Amazon Titan v2
   - Amazon Titan Multimodal Embeddings G1
   - Cohere Embed English
   - Cohere Embed Multilíngue
3. AWS OpenSearch coleção:
   - Você pode usar uma coleção de serviços provisionados ou sem servidor OpenSearch .
   - A coleção OpenSearch Service deve ter pelo menos um índice.
   - Se você planeja usar uma coleção OpenSearch sem servidor, certifique-se de criar uma coleção de pesquisa vetorial. Para obter detalhes sobre como configurar um índice vetorial, consulte Pré-requisitos para seu próprio armazenamento de vetores para uma base de conhecimento. Para saber mais sobre vetorização, consulte a explicação dos recursos do banco de dados vetoriais do Amazon OpenSearch Service.

> **Note**
>
> Ao criar um índice vetorial, você deve usar o nome do campo vetorial`embedded_data`.

- Se você planeja usar uma coleção OpenSearch provisionada, você precisa adicionar a função do aplicativo MSF (que contém a política de acesso do Opensearch) que foi criada pelo blueprint, como usuário principal à sua coleção. OpenSearch Além disso, confirme se a política de acesso OpenSearch está definida como “Permitir” ações. Isso é necessário para permitir um controle de acesso refinado.
- Opcionalmente, você pode ativar o acesso ao OpenSearch painel para visualizar os resultados. Consulte para ativar o controle de acesso refinado.

4. Faça login usando uma função que permite aws: CreateStack permissões.
5. Acesse o painel do console do MSF e selecione Criar aplicativo de streaming.
6. Em Escolha um método para configurar o aplicativo de processamento de stream, selecione Usar um Blueprint.
7. Selecione Plano de aplicativo de IA em tempo real no menu suspenso Planos.
8. Forneça as configurações desejadas. Consulte Criar configurações de página.
9. Selecione Implantar Blueprint para iniciar uma CloudFormation implantação.
10. Quando a CloudFormation implantação estiver concluída, acesse o aplicativo Flink implantado. Verifique as propriedades de tempo de execução do aplicativo.
11. Você pode optar por alterar/adicionar propriedades de tempo de execução ao seu aplicativo. Consulte Configuração de propriedades de tempo de execução para obter detalhes sobre como configurar essas propriedades.

> **Note**
>
> Nota:
> Se você estiver usando OpenSearch provisioned, certifique-se de habilitar o controle de acesso refinado.
> Se seu cluster provisionado for privado, adicione-o `https://` à URL do endpoint da VPC OpenSearch provisionada e altere `sink.os.endpoint` para apontar para esse endpoint.

Se seu cluster provisionado for público, certifique-se de que seu aplicativo MSF possa acessar a Internet. Para obter mais informações, consulte >>>>>> express-brokers-publication-merge type="documentation" url="managed- flink/latest/java/vpc - internet.html ">Acesso à Internet e aos serviços de um serviço gerenciado conectado à VPC para o aplicativo Apache Flink.

12. Quando estiver satisfeito com todas as configurações, selecioneRun. O aplicativo começará a ser executado.
13. Bombeie mensagens em seu cluster MSK.
14. Navegue até o cluster do Opensearch e acesse o OpenSearch painel.
15. No painel, selecione Descobrir no menu à esquerda. Você deve ver documentos persistentes junto com suas incorporações vetoriais.
16. Consulte Trabalhando com coleções de pesquisa vetorial para ver como você pode usar os vetores armazenados no índice.

Criar configurações de página

Este tópico descreve a criação de configurações de página a serem consultadas ao especificar configurações para blueprints de aplicativos de IA em tempo real.

Nome da aplicação

Campo existente no MSF, dê qualquer nome ao seu aplicativo.

Cluster do MSK

Selecione na lista suspensa o cluster MSK que você criou durante a configuração.

Tópicos

Adicione o nome do (s) tópico (s) que você criou na configuração.

Tipo de dados do fluxo de entrada

Escolha String se você fornecer entrada de string para o fluxo MSK.

Escolha JSON se a entrada no fluxo MSK for JSON. Em chaves JSON incorporadas, escreva os nomes dos campos em seu JSON de entrada cujo valor você deseja enviar ao Bedrock para gerar incorporações.

Modelo de incorporação Bedrock

Selecione um na lista. Certifique-se de ter acesso ao modelo escolhido, caso contrário, a pilha poderá falhar. Consulte Adicionar ou remover o acesso aos modelos de base do Amazon Bedrock.

OpenSearch agrupamento

Selecione no cluster que você criou no menu suspenso.

OpenSearch nome do índice de vetores

Selecione o índice vetorial que você criou na etapa acima.

Cota do Amazon MSK

Você Conta da AWS tem cotas padrão para o Amazon MSK. Salvo indicação em contrário, cada cota por conta é específica da região dentro da sua. Conta da AWS

Tópicos

Solicitando um aumento de cota no Amazon MSK

Você pode solicitar um aumento de cota para cada região usando o console Service Quotas ou um AWS CLI caso de suporte. Se uma cota ajustável não estiver disponível no console de Cotas de Serviço, use AWS Support Center Console o para criar um caso de aumento da cota de serviço.

O Support pode aprovar, negar ou aprovar parcialmente suas solicitações de aumento de cota. Os aumentos não são concedidos imediatamente e podem levar alguns dias para entrar em vigor.

Para solicitar um aumento, visite o Console do Service Quotas

1. Abra o console do Service Quotas em https://console.aws.amazon.com/servicequotas/.
2. Na barra de navegação, na parte superior da tela, selecione uma região.
3. No painel de navegação à esquerda, selecione Serviços da AWS.
4. Na caixa Localizar serviços, digite e, em seguida**`msk`**, escolha Amazon Managed Streaming for Apache Kafka (MSK).
5. Em Cotas de serviço, escolha o nome da cota para a qual você deseja solicitar um aumento. Por exemplo, .**`Number of brokers per account`**
6. Escolha Solicitar aumento no nível da conta.
7. Em Aumentar valor da cota, insira um novo valor da cota.

8. Escolha Solicitar.
9. (Opcional) Para visualizar qualquer solicitação pendente ou resolvida recentemente no console, escolha Painel no painel de navegação esquerdo. Para solicitações pendentes, escolha o status da solicitação para abrir o recibo da solicitação. O status inicial de uma solicitação é Pending (Pendente). Depois que o status mudar para Cota solicitada, você verá o número do caso no Support. Escolha o número do caso para abrir o tíquete de sua solicitação.

Para obter mais informações, incluindo como usar AWS CLI ou solicitar um aumento SDKs de cota, consulte Solicitando um aumento de cota no Guia do Usuário de Quotas de Serviço.

Cota de corretor Amazon MSK Standard

A tabela a seguir descreve as cotas para corretores Standard.

Dimensão	Cota	Observações
Corretores por conta	90	Para solicitar uma cota maior, acesse o console do Service Quotas.
Corretores por cluster	30 para clusters ZooKeeper baseados 60 para clusters KRaft baseados	Para solicitar uma cota maior, acesse o console do Service Quotas.
Armazenamento mínimo por corretor	15 GiB	
Armazenamento máximo por corretor	16384 GiB	
Máximo de conexões TCP por agente (controle de acesso IAM)	3000	Para aumentar esse limite, você pode ajustar a propriedade de `listener.name.client_iam_public.max.connections` configuração `listener.name.client_iam.ma`

Dimensão	Cota	Observações
		`x.connections` ou usando a `AlterConfig` API Kafka ou a `kafka-configs.sh` ferramenta. É importante observar que aumentar qualquer propriedade para um valor alto pode resultar em indisponibilidade.
Taxa máxima de conexões TCP por agente (IAM)	100 por segundo (tamanhos de instância M5 e M7g) 4 por segundo (tamanho de instância t3)	Para processar novas tentativas em conexões com falha, você pode definir o parâmetro de configuração `reconnect.backoff.ms` no lado do cliente. Por exemplo, se você quiser que um cliente tente novamente as conexões após 1 segundo, `reconnect.backoff.ms` defina `1000` como. Para obter mais informações, consulte reconnect.backoff.ms na documentação do Apache Kafka.
Máximo de conexões TCP por agente (não IAM)	N/D	O MSK não impõe limites de conexão para autenticação não IAM. Você deve monitorar outras métricas, como uso de CPU e memória, para garantir que não sobrecarregue seu cluster devido ao excesso de conexões.

Dimensão	Cota	Observações
Configurações por conta da	100	Para solicitar uma cota maior, acesse o console do Service Quotas. Para atualizar a configuração ou a versão do Apache Kafka de um cluster do MSK, primeiro certifique-se de que o número de partições por agente esteja abaixo dos limites descritos em Dimensione seu cluster adequadamente: número de partições por agente padrão.
Revisões de configuração por conta	50	

Cota do agente Amazon MSK Express

A tabela a seguir descreve as cotas para corretores Express.

Dimensão	Cota	Observações
Corretores por conta	90	Para solicitar uma cota maior, acesse o console do Service Quotas.
Corretores por cluster	30	Para solicitar uma cota maior, acesse o console do Service Quotas.
Armazenamento máximo	Ilimitado	

Dimensão	Cota	Observações
Máximo de conexões TCP por agente (controle de acesso IAM)	3000	Para aumentar o limite de conexão, ajuste uma das seguintes propriedades de configuração usando a AlterConfig API Kafka ou a ferramenta kafka-configs.sh: • `listener.name.client_iam.max.connections` • `listener.name.client_iam_public.max.connections` Definir essas propriedades com um valor alto pode resultar na indisponibilidade do cluster.
Taxa máxima de conexões TCP por agente (IAM)	100 por segundo	Para processar novas tentativas em conexões com falha, você pode definir o parâmetro de configuração `reconnect.backoff.ms` no lado do cliente. Por exemplo, se você quiser que um cliente tente novamente as conexões após 1 segundo, `reconnect.backoff.ms` defina `1000` como. Para obter mais informações, consulte reconnect.backoff.ms na documentação do Apache Kafka.

Dimensão	Cota	Observações
Máximo de conexões TCP por agente (não IAM)	N/D	O MSK não impõe limites de conexão para autentica ção não IAM. No entanto, você deve monitorar outras métricas, como uso de CPU e memória, para garantir que não sobrecarregue seu cluster devido ao excesso de conexões.
Configurações por conta da	100	Para solicitar uma cota maior, acesse o console do Service Quotas. Para atualizar a configuração ou a versão do Apache Kafka de um cluster do MSK, primeiro certifiqu e-se de que o número de partições por agente esteja abaixo dos limites descritos em Dimensione seu cluster corretamente: número de partições por agente Express.
Revisões de configuração por conta	50	
Entrada máxima por corretora	Recomendado: 15,6 - 500,0 MBps	Com base no tamanho da instância.
Saída máxima por corretora	Recomendado: 31,2 - 1000,0 MBps	Com base no tamanho da instância.

Tópicos

- Limites de aceleração da taxa de transferência da corretora expressa por tamanho da corretora
- Cota de partição do Express Broker

Limites de aceleração da taxa de transferência da corretora expressa por tamanho da corretora

A tabela a seguir lista o limite máximo e recomendado do acelerador de produtividade relacionado à entrada e saída para diferentes tamanhos de corretores. Nesta tabela, a taxa de transferência recomendada é representada como Desempenho sustentado, que é o limite até o qual seus aplicativos não sofrerão nenhuma degradação de desempenho. Se você operar além desses limites em qualquer dimensão, poderá obter mais produtividade, mas também poderá sofrer uma degradação do desempenho. A cota máxima é o limite no qual seu cluster limitará o tráfego. read/write Seus aplicativos não poderão operar além desse limite.

Tamanho da instância	Desempenho sustentado (MBps) para entrada	Cota máxima (MBps) para entrada	Desempenho sustentado (MBps) para saída	Cota máxima (MBps) para saída
express.m7g.large	15,6	23,4	31.2	58,5
express.m7g.xlarge	31.2	46,8	62.5	117
express.m7g.2xlarge	62.5	93,7	125	234.2
express.m7g.4xlarge	124,9	187,5	249,8	468,7
express.m7g.8xlarge	250	375	500	937,5
express.m7g.12xlarge	375	562,5	750	1406,2
express.m7g.16xlarge	500	750	1000	1875

Cota de partição do Express Broker

A tabela a seguir mostra o número recomendado de partições (incluindo réplicas líder e seguidora) para cada agente Express. Você não pode exceder o número máximo de partições mencionado na tabela a seguir para cada corretor Express.

Para obter informações sobre as melhores práticas a serem consideradas ao atribuir partições aos corretores Express, consulte. Dimensione seu cluster corretamente: número de partições por agente Express

Tamanho do agente	Número recomendado de partições (incluindo partições líderes e seguidoras) por agente	Número máximo de partições por agente
`express.m7g.large`	1000	1500
`express.m7g.xlarge`	1000	2000
`express.m7g.2xlarge`	2500	4000
`express.m7g.4xlarge`	6000	8000
`express.m7g.8xlarge`	12000	16000
`express.m7g.12xlarge`	16000	24000
`express.m7g.16xlarge`	20000	32000

Cotas do replicador do MSK

- Máximo de 15 replicadores do MSK por conta.
- O Replicador do MSK replica somente até 750 tópicos de forma ordenada. Se você precisar replicar mais tópicos, recomendamos criar um replicador separado. Acesse o console do Service Quotas se precisar de suporte para mais de 750 tópicos por replicador. Você pode monitorar o número de tópicos que estão sendo replicados usando a métrica TopicCount "".
- Um throughput máximo de entrada de 1 GB por segundo por replicador do MSK. Solicite uma cota maior acessando o console do Service Quotas.

- Tamanho do registro do Replicador do MSK: um tamanho máximo de registro de 10 MB (message.max.bytes). Solicite uma cota maior acessando o console do Service Quotas.

Cota do MSK Serverless

As cotas especificadas na tabela a seguir são por cluster, salvo indicação em contrário.

> **Note**
>
> Caso tenha algum problema com os limites da cota de serviço, crie um caso de suporte com seu caso de uso e o limite solicitado.

Dimensão	Quota	Resultado de violação de cota
Throughput máximo de entrada	200 MBps	Desaceleração com duração de controle de utilização em resposta
Throughput máximo de saída	400 MBps	Desaceleração com duração de controle de utilização em resposta
Duração máxima de retenção	Ilimitado	N/D
Número máximo de conexões de cliente	3000	Fechamento da conexão
Máximo de tentativas de conexão	100 por segundo	Fechamento da conexão
Tamanho máximo de mensagem	8 MiB	A solicitação falha com ErrorCode: INVALID_REQUEST
Taxa máxima de solicitação	15.000 por segundo	Desaceleração com duração de controle de utilização em resposta

Dimensão	Quota	Resultado de violação de cota
Taxa máxima de APIs solicitações de gerenciamento de tópicos	2 por segundo	Desaceleração com duração de controle de utilização em resposta
Máximo de bytes de busca por solicitação	55 MB	A solicitação falha com ErrorCode: INVALID_REQUEST
Número máximo de grupos de consumidores	500	JoinGroup falha na solicitação
Número máximo de partições (líderes)	2.400 para tópicos não compactados, 120 para tópicos compactados. Para solicitar um ajuste de cota de serviço, crie um caso de suporte com seu caso de uso e limite solicitado.	A solicitação falha com ErrorCode: INVALID_REQUEST
Taxa máxima de criação e exclusão de partições	250 em 5 minutos	A solicitação falha com ErrorCode: THROUGHPUT_QUOTA_EXCEEDED
Throughput máximo de entrada por partição	5 MBps	Desaceleração com duração de controle de utilização em resposta
Throughput máximo de saída por partição	10 MBps	Desaceleração com duração de controle de utilização em resposta
Tamanho máximo da partição (para tópicos compactados)	250 GB	A solicitação falha com ErrorCode: THROUGHPUT_QUOTA_EXCEEDED
Número máximo de clientes VPCs por cluster sem servidor	5	

Dimensão	Quota	Resultado de violação de cota
Número máximo de clusters com tecnologia sem servidor por conta	10. Para solicitar um ajuste de cota de serviço, crie um caso de suporte com seu caso de uso e limite solicitado.	

Cota do MSK Connect

- Até 100 plug-ins personalizados.
- Até 100 configurações de operador.
- Até 60 operadores conectados. Se um conector estiver configurado com capacidade de ajuste de escala automático, o número máximo de operadores que o conector está configurado para ter é o número que o MSK Connect usa para calcular a cota da conta.
- Até 10 operadores por conector.

Para solicitar uma cota maior para o MSK Connect, acesse o console do Service Quotas.

Histórico do documento para o Guia do desenvolvedor do Amazon MSK

A tabela a seguir descreve as alterações importantes feitas no Guia do desenvolvedor do Amazon MSK.

Última atualização da documentação: 25 de junho de 2024

Alteração	Descrição	Data
StorageUsed métrica para corretores Express	O Amazon MSK agora inclui uma nova métrica de nível DEFAULT, `StorageUsed` para corretores Express, que fornece visibilidade em nível de cluster do consumo total de armazenamento, excluindo réplicas. Para obter mais informações, consulte Monitoramento de nível DEFAULT para corretores Express.	2025-07-24
Alto número de partições para corretores Amazon MSK Express	A Amazon MSK lançou uma alta contagem de partições para corretores Express. Para obter mais informações, consulte Cota de partição do Express Broker.	2025-07-21
Novas métricas do Amazon MSK Connect	O MSK Connect adicionou duas novas métricas — `SinkConsumerByteRate` e `SourceProducerByteRate` para medir as taxas de transferência de dados	2025-06-30

Alteração	Descrição	Data
	do conector. Para obter mais informações, consulte Monitorando o MSK Connect.	
Visão geral dos tópicos de introdução do MSK Provisioned	Reestruturou completamente os tópicos de introdução do MSK Provisioned para melhorar a experiência do usuário, o fluxo de conteúdo e a legibilidade. Os tópicos revisados também incluem documentação detalhada de todas as opções de console, incluindo modos de armazenamento, métodos de autenticação e níveis de monitoramento com orientações claras para decisões. Para obter mais informações, consulte Comece a usar o Amazon MSK.	2025-06-28
Support para Express broker no Apache Kafka versão 3.8.x	O Amazon MSK agora oferece suporte a corretores Express no Apache Kafka versão 3.8.x. Para obter mais informações, consulte os corretores do Amazon MSK Express.	2025-06-05

Alteração	Descrição	Data
Novas informações de solução de problemas do Amazon MSK Replicator	O Guia do Desenvolvedor do Amazon Managed Streaming for Apache Kafka agora inclui documentação abrangent e de solução de problemas do MSK Replicator com script de diagnóstico e procedimentos detalhados de mitigação de erros. Para obter mais informações, consulte Solucionar problemas de falhas do MSK Replicator usando métricas. Replicato rFailure	2025-05-09
Renovações de certificados sem interrupções	A Amazon MSK lançou renovações de certifica dos sem interrupções para corretores Express a fim de eliminar o tempo de inatividade para manutenção durante as atualizações obrigatórias de 13 meses do certificado. Para obter mais informaçõ es, consulte Criptografia do Amazon MSK.	2025-05-05
Support para Apache Kafka versão 4.0.x	O Amazon MSK agora oferece suporte ao Apache Kafka versão 4.0.x. Para obter mais informações, consulte Versões suportadas do Apache Kafka.	2025-05-02

Alteração	Descrição	Data
Lançamento do Amazon MSK Connect nas regiões da China	O MSK Connect agora está disponível em todas as regiões da China — China (Pequim) e China (Ningxia).	2025-04-10
Support for Apache Kafka versão 3.9.x	O Amazon MSK agora oferece suporte ao Apache Kafka versão 3.9.x. Para obter informações, consulte Versões suportadas do Apache Kafka.	2025-04-21
Conector de coletor Amazon EventBridge Kafka para MSK Connect	O Guia do Desenvolvedor do Amazon Managed Streaming for Apache Kafka agora inclui um tópico abrangent e que descreve como usar o conector de coletor Kafka EventBridge com o MSK Connect. Para obter mais informações, consulte Configurar o conector de coletor EventBridge Kafka para o MSK Connect.	2025-03-28
Atualização de cotas de corretores expressos	O Guia do Desenvolvedor do Amazon Managed Streaming for Apache Kafka agora inclui informações sobre limites de taxa de transferência para entrada e saída para corretore s Express. Para obter mais informações, consulte a cota do agente Amazon MSK Express.	2025-03-06

Alteração	Descrição	Data
Support para Apache Kafka versão 3.8.x	O Amazon MSK agora oferece suporte ao Apache Kafka versão 3.8.x. Para obter mais informações, consulte Versões suportadas do Apache Kafka.	2025-02-20
Data de fim do suporte do Amazon MSK versão 3.4.0 revisada	A data revisada de fim do suporte para a versão 3.4.0 do Apache Kafka é 4 de agosto de 2025. Para obter mais informações, consulte Versões suportadas do Apache Kafka.	2025-02-18
UpdateConnector Lançamento da API para modificar as configurações existentes do conector MSK Connect	O Amazon MSK agora inclui a UpdateConnectorAPI para modificar as configurações existentes do conector MSK Connect, eliminando a necessidade de criar novos conectores. Além disso, foi adicionada documentação sobre DescribeConnectorOperatione ListConnectorOperations APIs para rastrear as operações de atualização do conector e manter trilhas de auditoria históricas das alterações de configuração.	2025-01-12

Alteração	Descrição	Data
AWS PrivateLink documentação de endpoints de VPC da interface	O guia do desenvolvedor do Amazon Managed Streaming for Apache Kafka agora AWS PrivateLink inclui documentação de endpoints de VPC de interface. Para obter mais informações, consulte Usar o Amazon MSK APIs com endpoints de interface VPC.	2024-12-18
Lançamento da versão 3.7.x do MSK Connect	O MSK Connect agora oferece suporte à versão 3.7.x. Para obter mais informações, consulte Compreender o MSK Connect.	2024-12-18
Foi adicionado o recurso de corretor expresso. Tópicos do Guia do Desenvolvedor reorganizados.	A MSK oferece suporte a corretores Standard e New Express.	2024-11-6
Adicionado o recurso de atualização do Graviton no local.	Você pode atualizar o tamanho do agende de cluster de M5 ou T3 para M7g ou de M7g para M5.	25/06/2024
Anunciada data do fim do suporte da versão 3.4.0.	A data do fim do suporte para o Apache Kafka versão 3.4.0 é 17 de junho de 2025.	2024-6-24

Alteração	Descrição	Data
Adicionado recurso de remoção de agente.	Você pode reduzir a capacidade de armazenamento e computação do cluster provisionado removendo conjuntos de agentes, sem impacto na disponibilidade, risco de durabilidade de dados ou interrupção nas aplicações de fluxo de dados.	16/05/2024
`WriteDataIdempotently` adicionado ao AWSMSKReplicator Execution Role	WriteDataIdempotently a permissão é adicionada à AWSMSKReplicator Execution Role política para oferecer suporte à replicação de dados entre clusters MSK.	16/05/2024
Agentes M7g do Graviton lançados no Brasil e em Bahrein.	O Amazon MSK agora oferece suporte à disponibilidade de corretores M7g nas regiões da América do Sul (sa-east-1, São Paulo) e Oriente Médio (me-south-1, Bahrein) usando processadores Graviton (processadores personalizados baseados em ARM criados pela Amazon Web Services). AWS	2024-2-07

Alteração	Descrição	Data
Versão dos agentes M7g do Graviton para a região da China	O Amazon MSK agora oferece suporte à disponibilidade de corretores m7G na região da China usando processadores AWS Graviton (processadores personalizados baseados em ARM criados pela Amazon Web Services).	2024-01-11
Política de suporte da versão do Amazon MSK Kafka	Foi adicionada uma explicação sobre a política de suporte da versão Kafka compatível com o Amazon MSK. Para obter mais informações, consulte Versões do Apache Kafka.	2023-12-08
Nova política de perfil de execução do serviço para compatibilidade com o Replicador do Amazon MSK.	O Amazon MSK adicionou uma nova política `AWSMSKReplicatorExecutionRole` para suporte ao Replicador do Amazon MSK. Para obter mais informações, consulte Políticas gerenciadas pela AWS : `AWSMSKReplicatorExecutionRole`.	2023-12-06
Suporte do M7g Graviton	O Amazon MSK agora oferece suporte a corretores M7g usando processadores AWS Graviton (processadores personalizados baseados em ARM criados pela Amazon Web Services).	2023-11-27

Alteração	Descrição	Data
Replicador do Amazon MSK	O replicador do Amazon MSK é um novo recurso que você pode usar para replicar dados entre clusters do Amazon MSK. O Amazon MSK Replicator inclui uma atualização da política do Amazon MSKFull Access. Para obter mais informações, consulte Políticas gerenciadas pela AWS : `AmazonMSKFullAccess`.	2023-09-28
Atualização com as práticas recomendadas do IAM.	Guia atualizado para alinhamento com as práticas recomendadas do IAM. Para obter mais informações, consulte Práticas recomendadas de segurança no IAM.	2023-03-08

Alteração	Descrição	Data
Atualizações ao perfil vinculado a serviço para compatibilidade com conectividade privada multi-VPC	O Amazon MSK agora inclui atualizações de funções AWSService RoleForKafka vinculadas a serviços para gerenciar interfaces de rede e endpoints de VPC em sua conta, tornando os agentes de cluster acessíveis aos clientes em sua VPC. O Amazon MSK usa permissões para `DescribeVpcEndpoints` , `ModifyVpcEndpoint` e `DeleteVpcEndpoints` . Para obter mais informações, consulte Perfis vinculados ao serviço para o Amazon MSK.	2023-03-08
Compatibilidade com Apache Kafka 2.7.2	Agora, o Amazon MSK é compatível com o Apache Kafka versão 2.7.2. Para obter mais informações, consulte Versões compatíveis do Apache Kafka.	2021-12-21
Compatibilidade com Apache Kafka 2.6.3	Agora, o Amazon MSK é compatível com o Apache Kafka versão 2.6.3. Para obter mais informações, consulte Versões compatíveis do Apache Kafka.	2021-12-21

Alteração	Descrição	Data
Pré-lançamento do MSK Serverless	O MSK Serverless é um novo recurso que você pode usar para criar clusters com a tecnologia sem servidor. Para obter mais informações, consulte MSK Serverless.	2021-11-29
Compatibilidade com Apache Kafka 2.8.1	Agora, o Amazon MSK é compatível com o Apache Kafka versão 2.8.1. Para obter mais informações, consulte Versões compatíveis do Apache Kafka.	2021-09-30
MSK Connect	O MSK Connect é um novo recurso que você pode usar para criar e gerenciar conectores do Apache Kafka. Para obter mais informações, consulte Saiba mais sobre o MSK Connect.	2021-09-16
Compatibilidade com Apache Kafka 2.7.1	Agora, o Amazon MSK é compatível com o Apache Kafka versão 2.7.1. Para obter mais informações, consulte Versões compatíveis do Apache Kafka.	2021-05-25
Compatibilidade com Apache Kafka 2.8.0	Agora, o Amazon MSK é compatível com o Apache Kafka versão 2.8.0. Para obter mais informações, consulte Versões compatíveis do Apache Kafka.	2021-04-28

Alteração	Descrição	Data
Compatibilidade com Apache Kafka 2.6.2	Agora, o Amazon MSK é compatível com o Apache Kafka versão 2.6.2. Para obter mais informações, consulte Versões compatíveis do Apache Kafka.	2021-04-28
Compatibilidade com atualização do tipo de agente	Agora, você pode alterar o tipo de agente de um cluster existente. Para obter mais informações, consulte Atualizar o tamanho do agente de cluster do Amazon MSK.	2021-01-21
Compatibilidade com Apache Kafka 2.6.1	Agora, o Amazon MSK é compatível com o Apache Kafka versão 2.6.1. Para obter mais informações, consulte Versões compatíveis do Apache Kafka.	2021-01-19
Compatibilidade com Apache Kafka 2.7.0	Agora, o Amazon MSK é compatível com o Apache Kafka versão 2.7.0. Para obter mais informações, consulte Versões compatíveis do Apache Kafka.	2020-12-29

Alteração	Descrição	Data
Não há novos clusters no Apache Kafka versão 1.1.1	Você não pode mais criar um novo cluster do Amazon MSK com o Apache Kafka versão 1.1.1. No entanto, se você tiver clusters existente s do MSK executando o Apache Kafka versão 1.1.1, poderá continuar usando todos os recursos atualment e suportados nesses clusters existentes. Para obter mais informações, consulte Versões do Apache Kafka.	2020-11-24
Métricas de atraso do consumidor	Agora, o Amazon MSK fornece métricas que você pode usar para monitorar o atraso do consumidor. Para obter mais informações, consulte Monitore um cluster provisionado do Amazon MSK.	2020-11-23
Compatibilidade com Cruise Control	O Amazon MSK agora oferece suporte LinkedIn ao Cruise Control. Para obter mais informações, consulte Use o LinkedIn Cruise Control para Apache Kafka com o Amazon MSK.	2020-11-17

Alteração	Descrição	Data
Compatibilidade com Apache Kafka 2.6.0	Agora, o Amazon MSK é compatível com o Apache Kafka versão 2.6.0. Para obter mais informações, consulte Versões compatíveis do Apache Kafka.	2020-10-21
Compatibilidade com Apache Kafka 2.5.1	Agora, o Amazon MSK é compatível com o Apache Kafka versão 2.5.1. Com o Apache Kafka versão 2.5.1, o Amazon MSK oferece suporte à criptografia em trânsito entre clientes e endpoints. ZooKeeper Para obter mais informações, consulte Versões compatíveis do Apache Kafka.	2020-09-30
Expansão automática de aplicação	Você pode configurar o Amazon Managed Streaming for Apache Kafka para expandir automaticamente o armazenamento do seu cluster em resposta ao aumento do uso. Para obter mais informações, consulte Escalabilidade automática para clusters.	2020-09-30

Alteração	Descrição	Data
Compatibilidade com segurança de nome de usuário e senha	Agora, o Amazon MSK é compatível com login em clusters usando nome de usuário e senha. O Amazon MSK armazena as credencia is no AWS Secrets Manager. Para obter mais informações, consulte Autenticação SASL/ SCRAM.	2020-09-17
Compatibilidade com a atualização da versão do Apache Kafka de um cluster do Amazon MSK	Agora, é possível atualizar a versão do Apache Kafka de um cluster existente do MSK.	28-05-2020
Suporte para nós de agente T3.small	O Amazon MSK agora oferece suporte à criação de clusters com corretores do EC2 tipo Amazon T3.small.	2020-04-08
Compatibilidade com Apache Kafka 2.4.1	Agora, o Amazon MSK é compatível com o Apache Kafka versão 2.4.1.	02-04-2020
Suporte para logs de agente do streaming	Agora, o Amazon MSK pode transmitir registros do broker para CloudWatch Logs, Amazon S3 e Amazon Data Firehose. O Firehose pode, por sua vez, entregar esses registros aos destinos que ele suporta, como OpenSearch o Service.	25-02-2020

Alteração	Descrição	Data
Compatibilidade com Apache Kafka 2.3.1	Agora, o Amazon MSK é compatível com o Apache Kafka versão 2.3.1.	19-12-2019
Monitoramento aberto	Agora, o Amazon MSK é compatível com monitoramento aberto usando o Prometheus.	04-12-2019
Compatibilidade com Apache Kafka 2.2.1	Agora, o Amazon MSK é compatível com o Apache Kafka versão 2.2.1.	31-07-2019
Disponibilidade geral	Os novos recursos incluem suporte ao uso de tags, autenticação, criptografia TLS, configurações e a capacidade de atualizar o armazenamento de agentes.	30-05-2019
Compatibilidade com Apache Kafka 2.1.0	Agora, o Amazon MSK é compatível com o Apache Kafka versão 2.1.0.	05-02-2019

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.