aws

开发人员指南

Amazon Managed Streaming for Apache Kafka

Amazon Managed Streaming for Apache Kafka: 开发人员指南

Table of Contents

欢迎使用《Amazon MSK 开发人员指南》

欢迎使用《Amazon MSK 开发人员指南》。以下主题可帮助您根据自己的需求开始使用本指南。

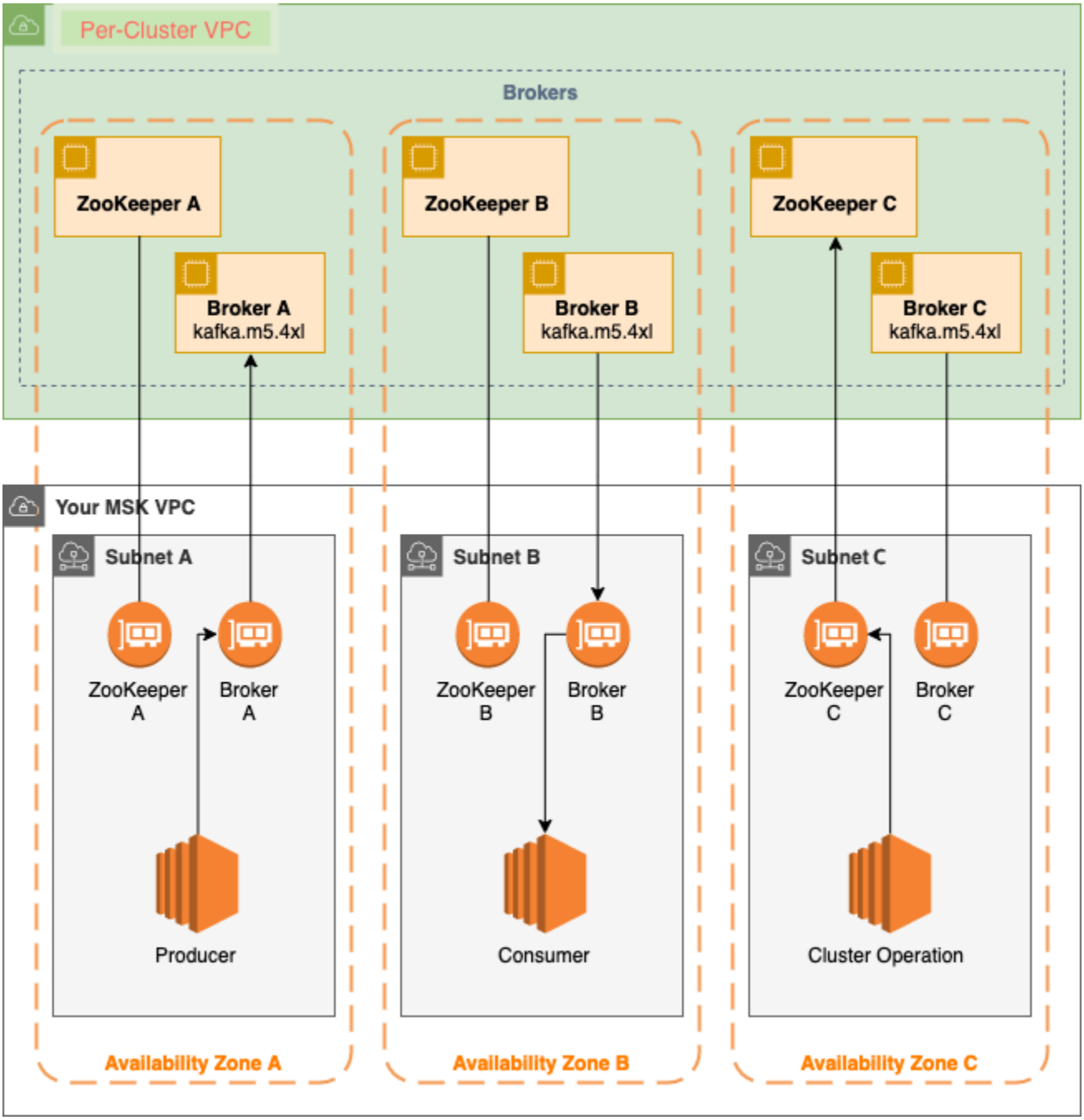
- 按照 开始使用 Amazon MSK 教程创建 Amazon MSK 集群。
- 深入了解 Amazon MSK 的工作原理 中的 Amazon MSK 的功能。
- 使用 MSK Serverless 无需管理和扩展集群容量即可运行 Apache Kafka。
- 用于 MSK Connect 将数据流入和流出 Apache Kafka 集群。

有关亮点、产品详细信息和定价，请参阅 Amazon MSK 服务页面。

什么是 Amazon MSK？

Amazon Managed Streaming for Apache Kafka（Amazon MSK）是一项完全托管式服务，让您能够构建并运行使用 Apache Kafka 来处理串流数据的应用程序。Amazon MSK 提供控制面板操作，例如，用于创建、更新和删除集群的操作。它允许您使用 Apache Kafka 数据层面操作，例如，用于生成和使用数据的操作。它运行 Apache Kafka 的开源版本。这意味着支持来自合作伙伴和 Apache Kafka 社区的现有应用程序、工具和插件，而无需更改应用程序代码。您可以使用 Amazon MSK 创建使用 the section called “支持的 Apache Kafka 版本” 下列出的任何 Apache Kafka 版本的集群。

下图概述了 Amazon MSK 的工作原理。



该图演示了以下各个组件之间的交互：

- 代理节点 – 创建 Amazon MSK 集群时，您可以指定 Amazon MSK 要在每个可用区中创建的代理节点数。在此图显示的示例集群中，每个可用区有一个代理。每个可用区都有自己的 Virtual Private Cloud (VPC) 子网。
- ZooKeeper 节点 – Amazon MSK 还会为您创建 Apache ZooKeeper 节点。Apache ZooKeeper 是一个开源服务器，可实现高度可靠的分布式协调。

- 生成器、使用器和主题创建者 – Amazon MSK 允许您使用 Apache Kafka 数据面板操作来创建主题以及生成和使用数据。
- 集群操作 您可以使用 AWS Management Console、AWS Command Line Interface（AWS CLI）或 SDK 中的 API 来执行控制面板操作。例如，您可以创建或删除 Amazon MSK 集群、列出账户中的所有集群、查看集群的属性以及更新集群中代理的数量和类型。

Amazon MSK 会检测集群的最常见故障情况并自动进行恢复，以尽可能降低对生成器和使用器应用程序的影响，使它们能够继续执行写入和读取操作。当 Amazon MSK 检测到代理故障时，它会解决故障或用新的代理替换运行不正常或无法访问的代理。此外，如果可能，它会重用旧代理的存储来减少 Apache Kafka 需要复制的数据。可用性影响将仅限于 Amazon MSK 完成检测和恢复所需的时间。恢复后，生成器和使用器应用程序可以继续与发生故障前使用的相同代理 IP 地址进行通信。

设置 Amazon MSK

首次使用 Amazon MSK 前，请完成以下任务：

任务

- 注册AWS
- 下载库和工具

注册AWS

当您注册 AWS 时，您的亚马逊云科技账户会自动注册 AWS 中的所有服务，包括 Amazon MSK。您只需为使用的服务付费。

如果您已有 AWS 账户，请跳到下一个任务。如果您还没有 AWS 账户，请使用以下步骤创建。

要注册亚马逊云科技账户

1. 打开 https://portal.aws.amazon.com/billing/signup。
2. 按照屏幕上的说明进行操作。

   在注册时，您将接到一通电话，要求您使用电话键盘输入一个验证码。

   当您注册 AWS 账户 时，系统将会创建一个 AWS 账户根用户。根用户有权访问该账户中的所有 AWS 服务和资源。作为安全最佳实践，请 为管理用户分配管理访问权限，并且只使用根用户执行需要根用户访问权限的任务。

下载库和工具

以下库和工具可帮助您使用 Amazon MSK：

- AWS Command Line Interface（AWS CLI）支持 Amazon MSK。AWS CLI 让您可以从命令行管理多个亚马逊云科技服务，并通过脚本自动执行这些服务。将您的 AWS CLI 升级到最新版本，确保其支持本用户指南中记录的 Amazon MSK 功能。有关如何升级 AWS CLI 的详细说明，请参阅安装 AWS Command Line Interface。安装之后 AWS CLI，必须对其进行配置。有关如何配置 AWS CLI 的更多信息，请参阅 aws configure。
- Amazon Managed Streaming for Kafka API Reference 记录了 Amazon MSK 支持的 API 操作。

- Amazon Web Services SDK for Go、Java、JavaScript、.NET、Node.js、PHP、Python 和 Ruby 包括 Amazon MSK 支持和示例。

开始使用 Amazon MSK

本教程举例说明如何执行以下操作：创建 MSK 集群、生成和使用数据以及使用指标来监控集群的运行状况。本示例并未提供您在创建 MSK 集群时可以选择的所有选项。为了简单起见，我们在本教程的各个部分均选择默认选项。这并不意味着它们是可用于设置 MSK 集群或客户端实例的唯一选项。

主题

步骤 1：创建 Amazon MSK 集群

在开始使用 Amazon MSK 的此步骤中，创建一个 Amazon MSK 集群。

要使用创建 Amazon MSK 集群 AWS Management Console

1. 登录并打开亚马逊 MSK 控制台，网址为 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。 AWS Management Console
2. 选择创建集群。
3. 对于创建方法，将快速创建选项保持为选中状态。快速创建选项允许您使用默认设置创建集群。
4. 对于集群名称，为您的集群输入一个描述性名称。例如，**`MSKTutorialCluster`**。
5. 对于常规集群属性，请选择已预置作为集群类型。
6. 从所有集群设置下的表中，复制并保存以下设置的值，稍后在本教程中会用到它们：
   - VPC
   - 子网
   - 与 VPC 关联的安全组
7. 选择创建集群。

8. 在集群摘要页面上，选中集群状态。在 Amazon MSK 预置集群时，状态从正在创建变为活动。当状态为活动时，您可连接到集群。有关集群状态的更多信息，请参阅 集群状态。

下一步

步骤 2：创建 IAM 角色

步骤 2：创建 IAM 角色

在此步骤中，您需执行两个任务。第一个任务是创建 IAM policy，以授予在集群上创建主题以及向这些主题发送数据的访问权限。第二个任务是创建 IAM 角色并将此策略与其关联。在后面的步骤中，您需创建代入此角色的客户端计算机，使用它在集群上创建主题并向该主题发送数据。

创建允许创建主题并写入主题的 IAM policy

1. 通过 https://console.aws.amazon.com/iam/ 打开 IAM 控制台。
2. 在导航窗格中，选择策略。
3. 选择创建策略。
4. 选择 JSON 选项卡，然后将编辑器窗口中的 JSON 替换为以下 JSON。

   将*##*替换为您创建集群的 AWS 区域的代码。将 *Account-ID* 替换为您的账户 ID。将 *MSKTutorialCluster* 替换为您的集群名称。

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

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

有关如何写入安全策略的说明，请参阅 the section called “IAM 访问控制”。

5. 选择下一步：标签。
6. 选择下一步：审核。
7. 为策略名称输入描述性名称，例如msk-tutorial-policy。
8. 选择 创建策略。

创建 IAM 角色并向其附加此策略

1. 在导航窗格中，选择角色。
2. 选择 创建角色。
3. 在常见用例下，选择 EC2，然后选择下一步：权限。
4. 在搜索框中，输入您之前为本教程创建的策略的名称。然后，选中策略左侧的复选框。
5. 选择下一步：标签。
6. 选择下一步：审核。
7. 在角色名称中，输入描述性名称，例如msk-tutorial-role。
8. 选择 创建角色。

下一步

步骤 3：创建客户端计算机

步骤 3：创建客户端计算机

在开始使用 Amazon MSK 的此步骤中，创建客户端计算机。可以使用此客户端计算机创建生成和使用数据的主题。为简单起见，您需在与 MSK 集群关联的 VPC 中创建此客户端计算机，以便客户端可以轻松连接到集群。

创建客户端计算机

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 选择 Launch instances。
3. 输入客户端计算机的名称，例如 **MSKTutorialClient**。
4. 对于亚马逊机器映像（AMI）类型，始终选中 Amazon Linux 2 AMI（HVM）– 内核 5.10，SSD 卷类型。
5. 保留 t2.micro 实例类型为选中状态。
6. 在密钥对（登录）下，选择创建新密钥对。为密钥对名称输入 **MSKKeyPair**，然后选择下载密钥对。此外，您还可使用现有密钥对。
7. 展开高级详细信息部分，然后选择您在步骤 2：创建 IAM 角色中创建的 IAM 角色。
8. 选择启动实例。
9. 选择查看实例。然后，在安全组列中，选择与新的实例关联的安全组。复制并保存安全组的 ID，以供稍后使用。
10. 通过以下网址打开 Amazon VPC 控制台：https://console.aws.amazon.com/vpc/。
11. 在导航窗格中，选择 Security Groups（安全组）。找到 ID 保存在 the section called “步骤 1：创建集群” 中的安全组。
12. 在入站规则选项卡上，选择编辑入站规则。
13. 选择 添加规则。
14. 在新规则中，选择类型列中的所有流量。在源列的第二个字段中，选择客户端计算机的安全组。这是您在启动客户端计算机实例后保存其名称的组。
15. 选择保存规则。现在，集群的安全组可以接受来自客户端计算机安全组的流量。

下一步

步骤 4：创建主题

步骤 4：创建主题

在开始使用 Amazon MSK 的此步骤中，您需在客户端计算机上安装 Apache Kafka 客户端库和工具，然后创建主题。

> **Warning**
>
> 本教程中使用的 Apache Kafka 版本号仅为示例。建议您使用与 MSK 集群版本相同的客户端版本。客户端版本较旧可能会缺少某些功能和关键错误修复。

查找 MSK 集群的版本

1. 转到 https://eu-west-2.console.aws.amazon.com/msk/
2. 选择 MSK 集群。
3. 请注意集群上所用 Apache Kafka 的版本。
4. 将本教程中的 Amazon MSK 版本号实例替换为在步骤 3 中获得的版本。

在客户端计算机上创建主题

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。然后选中您在 步骤 3：创建客户端计算机 中创建的客户端计算机名称旁边的复选框。
3. 选择 Actions (操作)，然后选择 Connect (连接)。按照控制台中的说明，连接到您的客户端计算机。
4. 通过运行以下命令在客户端计算机上安装 Java：

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

5. 运行以下命令以下载 Apache Kafka。

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

> **Note**
>
> 如果您希望使用此命令中使用的镜像站点之外的镜像站点，则可在 Apache 网站上选择其他镜像站点。

6. 在上一步中将 TAR 文件下载到的目录中运行以下命令。

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

7. 转到 `kafka_2.13-{YOUR MSK VERSION}/libs` 目录，然后运行以下命令以下载 Amazon MSK IAM JAR 文件。Amazon MSK IAM JAR 让客户端计算机可以访问集群。

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

8. 转到 `kafka_2.13-{YOUR MSK VERSION}/bin` 目录。复制以下属性设置并将其粘贴到新文件中。为文件 **`client.properties`** 命名并保存文件。

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

9. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
10. 等待集群的状态变为活动。这可能需要花几分钟的时间。在状态变为活动后，选择集群名称。这会将您引导至包含集群摘要的页面。
11. 选择查看客户端信息。
12. 复制私有端点的连接字符串。

    您将为每个代理获得三个端点。在以下步骤中，您只需要一个代理端点。
13. 运行以下命令，*`BootstrapServerString`*替换为您在上一步中获得的代理端点之一。

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

如果此命令成功，您将看到以下消息：`Created topic MSKTutorialTopic.`

下一步

步骤 5：生成和使用数据

步骤 5：生成和使用数据

在开始使用 Amazon MSK 的此步骤中，您需生成和使用数据。

生成和使用消息

1. 运行以下命令以启动控制台生成器。*BootstrapServerString*替换为您在创建主题中获得的纯文本连接字符串。有关如何检索此连接字符串的说明，请参阅 Getting the bootstrap brokers for an Amazon MSK cluster。

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

2. 输入所需的任何消息，然后按 Enter。重复执行此步骤两次或三次。每次输入一行并按 Enter 时，该行会作为单独的消息发送到您的 Apache Kafka 集群。
3. 将与客户端计算机的连接保持打开状态，然后在新窗口中打开与该计算机的第二个单独连接。
4. 在以下命令中，*BootstrapServerString*替换为之前保存的纯文本连接字符串。然后，要创建控制台使用器，使用客户端计算机的第二个连接运行以下命令。

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

   您开始看到之前使用控制台生成器命令时输入的消息。
5. 在生成器窗口中输入更多消息，并观察消息显示在使用器窗口中。

下一步

第 6 步：使用亚马逊 CloudWatch 查看亚马逊 MSK 指标

第 6 步：使用亚马逊 CloudWatch 查看亚马逊 MSK 指标

在 “开始使用亚马逊 MSK” 的这一步中，您将查看亚马逊中的亚马逊 MSK 指标。 CloudWatch

要在中查看 Amazon MSK 指标 CloudWatch

1. 打开 CloudWatch 控制台，网址为 https://console.aws.amazon.com/cloudwatch/。
2. 在导航窗格中，选择指标。
3. 选择所有指标选项卡，然后选择 AWS/Kafka。
4. 要查看代理级别的指标，请选择 Broker ID, Cluster Name (代理 ID，集群名称)。对于集群级别的指标，请选择 Cluster Name (集群名称)。
5. （可选）在图表窗格中，选择统计数据和时间段，然后使用这些设置创建 CloudWatch 警报。

下一步

步骤 7：删除为本教程创建的 AWS 资源

步骤 7：删除为本教程创建的 AWS 资源

在开始使用 Amazon MSK 的最后一步中，您需删除为本教程创建的 MSK 集群和客户端计算机。

要删除资源，请使用 AWS Management Console

1. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
2. 选择集群的名称。例如，MSK TutorialCluster。
3. 选择 Actions (操作)，然后选择 Delete (删除)。
4. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
5. 选择您为客户端计算机创建的实例，例如 **`MSKTutorialClient`**。
6. 选择实例状态，然后选择终止实例。

删除 IAM policy 和角色

1. 通过 https://console.aws.amazon.com/iam/ 打开 IAM 控制台。
2. 在导航窗格中，选择角色。
3. 在搜索框中，输入您为本教程创建的 IAM 角色的名称。
4. 选择角色 。然后选择删除角色并确认删除。
5. 在导航窗格中，选择策略。
6. 在搜索框中，输入您为本教程创建的策略的名称。

7. 选择策略，打开其摘要页面。在策略的摘要页面上，选择删除策略。
8. 选择 Delete （删除)。

Amazon MSK 的工作原理

Amazon MSK 集群是您可以在账户中创建的主要 Amazon MSK 资源。本节中的主题介绍如何执行常见的 Amazon MSK 操作。有关可以在 MSK 集群上执行的所有操作的列表，请参阅以下内容：

- 这些区域有：AWS Management Console
- Amazon MSK API Reference
- Amazon MSK CLI Command Reference

主题

创建 Amazon MSK 集群

> **Important**
>
> 创建集群之后便不能更改 Amazon MSK 集群的 VPC。

在创建 Amazon MSK 集群之前，您必须拥有一个 Amazon Virtual Private Cloud （VPC），并在该 VPC 内设置子网。

您需要在美国西部（北加利福尼亚）区域的两个不同可用区中使用两个子网。在提供 Amazon MSK 的其余区域中，您可以指定两到三个子网。您的子网必须位于不同的可用区中。在创建集群时，Amazon MSK 在您指定的子网之间平均分配代理节点。

代理类型

在创建 Amazon MSK 集群时，您可以指定其要使用的代理类型。Amazon MSK 支持以下代理类型：

- kafka.t3.small
- kafka.m5.large、kafka.m5.xlarge、kafka.m5.2xlarge、kafka.m5.4xlarge、kafka.m5.8xlarge、kafka.m5.12
- kafka.m7g.large、kafka.m7g.xlarge、kafka.m7g.2xlarge、kafka.m7g.4xlarge、kafka.m7g.8xlarge、kafka.

M7g 经纪商使用 G AWS raviton 处理器（由 Amazon Web Services 构建的基于 ARM 的定制处理器）。与同类的 M5 实例相比，M7g 经纪商的价格表现更高。M7g 代理比同类 M5 实例消耗的电量更少。MSK 在运行以下 Kafka 版本之一的集群上支持 m7g 代理：

- 2.8.2. 分层
- 3.3.2
- 3.4.0
- 3.5.1
- 3.6.0 带有分层存储

M7g 和 M5 代理的基准吞吐量性能比 T3 代理高，建议用于生产工作负载。M7g 和 M5 经纪商的每个代理也可以比 T3 经纪商拥有更多的分区。如果您正在运行较大的生产级工作负载或需要更多分区，请使用 M7g 或 M5 代理。要了解有关 M7g 和 M5 实例类型的更多信息，请参阅 Amazon EC2 通用实例。

T3 代理可以使用 CPU 积分来临时提高性能。如果您正在测试中小型流式处理工作负载，或者您的低吞吐量流式处理工作负载会临时出现吞吐量高峰，则可以使用 T3 代理进行低成本开发。我们建议您 proof-of-concept 进行测试，以确定 T3 代理是否足以应对生产或关键工作负载。要了解有关 T3 实例类型的更多信息，请参阅 Amazon EC2T3 Instances。

有关如何选择代理类型的更多信息，请参阅 最佳实践。

使用创建集群 AWS Management Console

1. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
2. 选择创建集群。
3. 指定集群的名称。
4. 在 VPC 列表中，选择要用于集群的 VPC。您还可以指定 Amazon MSK 创建集群要使用的 Apache Kafka 版本。2.8.1 版本是控制台中显示的默认版本，也是推荐的版本。
5. 如果您使用的是美国西部（北加利福尼亚）区域，请指定两个子网。在提供 Amazon MSK 的其他区域中，您可以指定两到三个子网。指定的子网必须位于不同的可用区中。
6. 选择所需配置类型。有关 Amazon MSK 配置的信息，请参阅 配置。
7. 指定您希望 MSK 在每个可用区中创建的代理类型和数量。每个可用区最少一个代理，每个集群最多 30 个代理。
8. （可选）为您的集群分配标签。标签是可选的。有关更多信息，请参阅 the section called “为集群添加标签”。
9. 您可以调整每个代理的存储量。创建集群后，您可以增加每个代理的存储量，但不能减少它。
10. 选择对传输中的数据进行加密所需的设置。默认情况下，Amazon MSK 会在集群中的代理之间传输数据时对数据进行加密。如果您希望在代理之间传输数据时不要对数据进行加密，请清除带有 Enable encryption within the cluster (在集群内启用加密) 标签的复选框。
11. 选择用于在客户端和代理之间传输数据时加密数据的三种设置之一。有关更多信息，请参阅 the section called “传输中加密”。
12. 选择要用于加密静态数据的 KMS 密钥类型。有关更多信息，请参阅 the section called “静态加密”。
13. 如果要验证客户端身份，请通过选择 Enable TLS client authentication (启用 TLS 客户端身份验证) 旁边的方框来选择该选项。有关身份验证的更多信息，请参阅 the section called “双向 TLS 身份验证”。
14. 选择所需的监控级别。此选择决定您获得的指标集。有关更多信息，请参阅 监控集群。
15. （可选）选择高级设置，然后选择自定义设置。您可以指定要向其授予对集群的访问权限的一个或多个安全组（例如，客户端计算机的安全组）。如果您指定与您共享的安全组，则必须确保您拥有对它们的权限。具体来说，您需要 `ec2:DescribeSecurityGroups` 权限。有关示例，请参阅 Amazon EC2：允许以编程方式和在控制台中管理与特定 VPC 关联的 EC2 安全组。
16. 选择创建集群。
17. 在集群摘要页面上，选中集群状态。在 Amazon MSK 预置集群时，状态从正在创建变为活动。当状态为活动时，您可连接到集群。有关集群状态的更多信息，请参阅 集群状态。

使用创建集群 AWS CLI

1. 复制以下 JSON 并将其保存到文件中。将文件命名为 brokernodegroupinfo.json。将 JSON 中的子网 ID 替换为与子网对应的值。这些子网必须位于不同的可用区中。将 *"Security-Group-ID"* 替换为客户端 VPC 的一个或多个安全组的 ID。与这些安全组关联的客户端可以访问集群。如果您指定与您共享的安全组，则必须确保您拥有对它们的权限。具体来说，您需要 ec2:DescribeSecurityGroups 权限。有关示例，请参阅 Amazon EC2：允许以编程方式和在控制台中管理与特定 VPC 关联的 EC2 安全组。最后，将更新后的 JSON 文件保存在已 AWS CLI 安装的计算机上。

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

> **Important**
>
> 如果您使用的是美国西部（北加利福尼亚）区域，请确切指定两个子网。对于提供 Amazon MSK 的其它区域，您可以指定两个或三个子网。指定的子网必须位于不同的可用区中。在创建集群时，Amazon MSK 在您指定的子网之间平均分配代理节点。

2. 在保存brokernodegroupinfo.json文件的目录中运行以下 AWS CLI 命令，将 *“Your-Cluster-Name”* 替换为您选择的名称。对于 *"Monitoring-Level"*，您可以指定以下三个值之一：DEFAULT、PER_BROKER 或 PER_TOPIC_PER_BROKER。有关这三个不同监控级别的信息，请参阅???。enhanced-monitoring 参数是可选的。如果未在 create-cluster 命令中指定该参数，监控级别即为 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"
```

该命令的输出如以下 JSON 所示：

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

> **Note**
>
> create-cluster 命令可能会返回错误，指示一个或多个子网所属的可用区不受支持。发生此种情况时，该错误会指示不受支持的可用区。请创建不使用不受支持的可用区的子网，然后重试 create-cluster 命令。

3. 保存 ClusterArn 键的值，因为您需要该键才能对集群执行其他操作。
4. 运行以下命令来检查集群的 STATE。在 Amazon MSK 预置集群时，STATE 值从 CREATING 变为 ACTIVE。当状态为 ACTIVE 时，您可连接到集群。有关集群状态的更多信息，请参阅 集群状态。

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

使用自定义 Amazon MSK 配置创建集群 AWS CLI

有关自定义 Amazon MSK 配置以及如何创建这些配置的信息，请参阅 配置。

1. 将以下 JSON 保存到文件中，并将 *configuration-arn* 替换为创建集群要使用的配置的 ARN。

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

2. 运行 create-cluster 命令并使用 configuration-info 选项指向您在上一步中保存的 JSON 文件。以下是示例。

```
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
```

以下是运行此命令后的成功响应示例。

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

使用 API 创建集群

要使用 API 创建集群，请参阅CreateCluster。

删除 Amazon MSK 集群

> **Note**
>
> 如果集群存在自动扩缩策略，建议您在删除集群之前移除该策略。有关更多信息，请参阅 自动扩缩。

使用删除集群 AWS Management Console

1. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
2. 选择要删除的 MSK 集群旁边的方框来选择该集群。
3. 选择删除角色，然后确认删除。

使用删除集群 AWS CLI

运行以下命令，*ClusterArn*替换为您在创建集群时获得的 Amazon 资源名称 (ARN)。如果您没有该集群的 ARN，可以通过列出所有集群来找到它。有关更多信息，请参阅 the section called “列出集群”。

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

使用 API 删除集群

要使用 API 删除集群，请参阅DeleteCluster。

获取亚马逊 MSK 集群的 Apache ZooKeeper 连接字符串

使用获取 Apache ZooKeeper 连接字符串 AWS Management Console

1. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
2. 该表显示了此账户下当前区域的所有集群。选择集群名称以查看其说明。
3. 在集群摘要页面上，选择查看客户端信息。这显示了引导程序代理以及 Apache ZooKeeper 连接字符串。

使用获取 Apache ZooKeeper 连接字符串 AWS CLI

1. 如果您不知道集群的 Amazon 资源名称 (ARN)，您可以通过列出您账户中的所有集群来找到它。有关更多信息，请参阅 the section called “列出集群”。
2. 要获取 Apache ZooKeeper 连接字符串以及有关集群的其他信息，请运行以下命令，*ClusterArn*替换为集群的 ARN。

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

该 `describe-cluster` 命令的输出如以下 JSON 示例所示。

```
{
    "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"
    }
}
```

上一 JSON 示例在 `describe-cluster` 命令输出中显示 `ZookeeperConnectString` 键。复制与此键对应的值，并保存它以用于在集群上创建主题。

> **Important**
>
> 您的 Amazon MSK 集群必须处于`ACTIVE`状态才能获取 Apache ZooKeeper 连接字符串。当集群仍处于 `CREATING` 状态时，`describe-cluster` 命令的输出不包含 `ZookeeperConnectString`。如果发生这种情况，请等待几分钟，然后在集群进入 `ACTIVE` 状态后再次运行 `describe-cluster`。

使用 API 获取 Apache ZooKeeper 连接字符串

要使用 API 获取 Apache ZooKeeper 连接字符串，请参阅DescribeCluster。

获取 Amazon MSK 集群的引导代理

使用获取引导程序代理 AWS Management Console

引导代理一词是指 Apache Kafka 客户端可以用作连接集群起点的代理的列表。此列表不一定包括集群中的所有代理。

1. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
2. 该表显示了此账户下当前区域的所有集群。选择集群名称以查看其说明。
3. 在集群摘要页面上，选择查看客户端信息。这显示了引导程序代理以及 Apache ZooKeeper 连接字符串。

使用获取引导程序代理 AWS CLI

运行以下命令，*ClusterArn*替换为您在创建集群时获得的 Amazon 资源名称 (ARN)。如果您没有该集群的 ARN，可以通过列出所有集群来找到它。有关更多信息，请参阅 the section called “列出集群”。

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

对于使用 the section called “IAM 访问控制” 的 MSK 集群，该命令的输出如以下 JSON 示例所示。

```
{
    "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"
}
```

以下示例显示了已打开公共访问的集群的引导代理。使用`BootstrapBrokerStringPublicSaslIam`用于公共访问，使用`BootstrapBrokerStringSaslIam`字符串进行内部访问 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"
}
```

引导代理字符串应包含来自部署 MSK 集群的可用区的三个代理（除非只有两个代理可用）。

使用 API 获取引导代理

要使用 API 获取引导程序代理，请参阅GetBootstrapBrokers。

列出 Amazon MSK 集群

使用列出集群 AWS Management Console

1. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
2. 该表显示了此账户下当前区域的所有集群。选择要查看其详细信息的集群的名称。

使用列出集群 AWS CLI

运行以下命令。

```
aws kafka list-clusters
```

使用 API 列出集群

要使用 API 列出集群，请参阅ListClusters。

存储管理

Amazon MSK 提供的功能可帮助您管理 MSK 集群的存储。

主题

- 分层存储
- 纵向扩展代理存储空间
- 预置存储吞吐量

分层存储

分层存储是 Amazon MSK 的低成本存储层，可扩展到几乎无限的存储空间，支持经济高效地构建流数据应用程序。

您可以创建配置有能平衡性能和成本的分层存储的 Amazon MSK 集群。Amazon MSK 将流数据存储在性能优化型主存储层中，直到数据达到 Apache Kafka 主题的保留期限。然后，Amazon MSK 会自动将数据移入新的低成本存储层。

当应用程序开始从分层存储中读取数据时，前几个字节的读取延迟可能会增加。开始按顺序从低成本层读取其余数据时，可能会出现与主存储层类似的延迟。您无需为低成本分层存储预置任何存储，也不需要管理基础设施。您可以存储任意数量的数据，但只需按实际用量付费。此功能与 KIP-405: Kafka Tiered Storage 中引入的 API 兼容。

以下是分层存储的一些功能：

- 您可以扩展到几乎无限的存储空间，不必猜测如何扩展 Apache Kafka 基础设施。
- 您可以在 Apache Kafka 主题中延长数据保留时间，也可以增加主题存储空间，不必增加代理数量。
- 其提供了持续时间更长的安全缓冲区来应对处理中的意外延迟。
- 您可以使用现有的流处理代码和 Kafka API 按确切的生产顺序重新处理旧数据。
- 分区重新平衡速度更快，因为二级存储上的数据不需要跨代理磁盘进行复制。
- 代理和分层存储之间的数据只会在 VPC 内移动，而不会通过互联网进行传输。
- 客户端计算机连接到启用了分层存储的新集群的过程，与连接到未启用分层存储的集群的过程相同。请参阅创建客户端计算机。

分层存储要求

- 您必须使用 Apache Kafka 客户端版本 3.0.0 或更高版本来创建启用了分层存储的新主题。要将现有主题过渡到分层存储，您可以重新配置使用低于 3.0.0 的 Kafka 客户端版本（支持的最低 Apache Kafka 版本为 2.8.2.tiered）的客户端计算机来启用分层存储。请参阅 步骤 4：创建主题。
- 启用了分层存储的 Amazon MSK 集群必须使用版本 3.6.0 或 2.8.2.tiered。

.

分层存储的约束和限制

分层存储存在以下约束和限制：

- 分层存储仅适用于预置模式集群。
- 分层存储不支持 t3.small 代理类型。
- 低成本存储的最短保留期为 3 天。主存储不存在最短保留期。
- 分层存储不支持代理上的多个日志目录（与 JBOD 相关的功能）。
- 分层存储不支持压缩主题。确保所有已开启分层存储的主题已将 cleanup.policy 配置为只能 “删除”。
- 可以为单个主题禁用分层存储，但不能禁用整个集群的分层存储。一旦禁用，就无法再为主题启用分层存储。
- 如果您使用 Amazon MSK 版本 2.8.2.tiers，则只能迁移到另一个支持分层存储的 Apache Kafka 版本。如果您不想继续使用支持分层存储的版本，请创建一个新的 MSK 集群并将数据迁移到该集群。
- 该 kafka-log-dirs 工具无法报告分层存储数据大小。该工具只会报告主存储中日志段的大小。

如何将日志段复制到分层存储

当您为新主题或现有主题启用分层存储时，Apache Kafka 会将已关闭的日志段从主存储复制到分层存储。

- Apache Kafka 仅复制已关闭的日志段。它将日志段中的所有消息复制到分层存储。
- 活动区段不符合分层条件。日志段大小 (segment.bytes) 或段滚动时间 (segment.ms) 控制数据段关闭的速率，以及 Apache Kafka 随后将段复制到分层存储的速率。

启用了分层存储的主题的保留设置与未启用分层存储的主题的保留设置不同。以下规则控制启用了分层存储的主题中消息的保留情况：

- 您可以在 Apache Kafka 中使用两个设置来定义保留期：log.retention.ms（时间）和 log.retention.bytes（大小）。这些设置决定了 Apache Kafka 在集群中保留的数据的总时长和总大小。无论是否启用分层存储模式，都需要在集群级设置这些配置。您可以使用主题配置覆盖主题级别的设置。
- 启用分层存储时，还可以指定高性能主存储层存储数据的时长。例如，如果主题的总体保留期 (log.retention.ms) 设置为 7 天，本地保留期 (local.retention.ms) 设置为 12 小时，则集群主存储仅保留前 12 小时的数据。低成本存储层可将数据保留整整 7 天。
- 一般的保留设置适用于完整日志。这包括其分层和主要部分。
- local.retention.ms 或 local.retention.bytes 设置控制消息在主存储中的保留情况。当完整日志的数据达到主存储保留设置阈值 (local.retention.ms/bytes) 时，Apache Kafka 会将主存储中的数据复制到分层存储。然后，数据就符合过期条件。

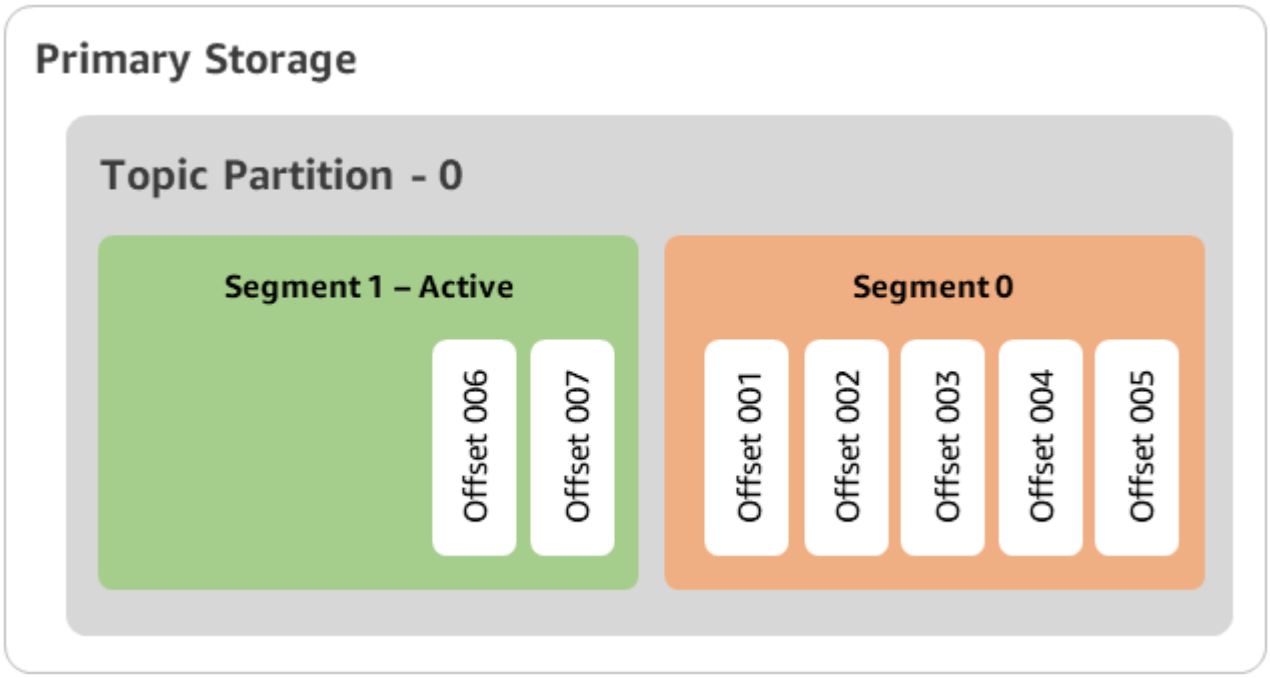
- 当 Apache Kafka 将日志段中的消息复制到分层存储时，会根据 retention.ms 或 retention.bytes 设置将该消息从集群中删除。

分层存储场景示例

此场景说明了启用分层存储后，主存储中包含消息的现有主题的行为方式。在将 remote.storage.enable 设置为 `true` 后，您可以启用本主题的分层存储。在此示例中，retention.ms 设置为 5 天，local.rete ntion.ms 设置为 2 天。以下是段过期时的事件序列。

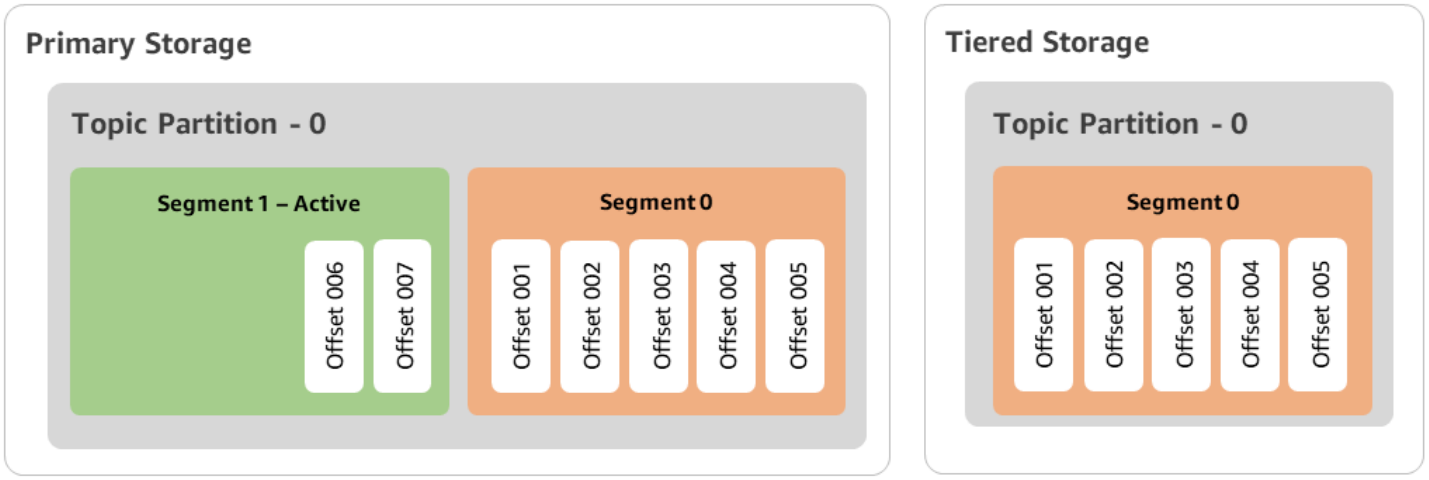
时间 T0 – 启用分层存储之前。

在为本主题启用分层存储之前有两个日志段。对于现有主题分区 0，其中一个日志段处于活动状态。



时间 T1（< 2 天）– 启用了分层存储。段 0 已复制到分层存储。

为本主题启用分层存储后，Apache Kafka 会在日志段满足初始保留设置后将段 0 复制到分层存储。Apache Kafka 还会保留段 0 的主存储副本。活动段 1 尚不满足复制到分层存储的条件。在此时间表中，Amazon MSK 尚未对段 0 和段 1 中的任何消息应用任何保留设置（local.retention.bytes/ms、retention.ms/bytes）。



时间 T2 – 本地保留设置生效。

两天后，Apache Kafka 复制到分层存储的段 0 的主保留设置将生效。这是因为 local.retention.ms 设置为了 2 天。主存储中的段 0 现已过期。活动段 1 既不满足过期要求，也符合复制到分层存储的条件。

时间 T3 – 总体保留设置生效。

五天后，保留设置生效，Kafka 会从分层存储中清除日志段 0 和关联消息。段 1 既不满足过期要求，也不符合复制到分层存储的条件，因为它处于活动状态。段 1 尚未关闭，因此不符合段滚动的条件。

使用分层存储创建带分层存储的 Amazon MSK 集群 AWS Management Console

1. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
2. 选择创建集群。
3. 为分层存储选择自定义创建。
4. 指定集群的名称。
5. 在集群类型中选择预置。
6. 为 Amazon MSK 选择支持分层存储的 Amazon Kafka 版本，用于创建集群。
7. 指定 kafka.t3.small 以外的代理类型。
8. 指定您希望 Amazon MSK 在每个可用区中创建的代理数量。每个可用区最少一个代理，每个集群最多 30 个代理。
9. 指定代理分布的可用区的数量。
10. 指定每个可用区部署的 Apache Kafka 代理的数量。
11. 选择存储选项。其中包括可启用分层存储模式的分层存储和 EBS 存储。
12. 按照集群创建向导中的剩余步骤操作。完成后，分层存储和 EBS 存储将作为集群存储模式显示在检查并创建视图中。
13. 选择 Create cluster（创建集群）。

使用分层存储创建带分层存储的 Amazon MSK 集群 AWS CLI

要在集群上启用分层存储，请使用正确的 Apache Kafka 版本和分层存储属性创建集群。请按照以下代码示例操作。此外，请完成下一节中的步骤，创建启用了分层存储的 Kafka 主题。

有关创建集群的支持属性的完整列表，请参阅 create-cluster。

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

创建启用了分层存储的 Kafka 主题

要完成在创建启用了分层存储的集群时开始的过程，您还要创建一个启用了分层存储的主题，其中包含后文代码示例中的属性。分层存储的专门属性如下：

- `local.retention.ms`（例如 10 分钟）为基于时间的保留设置，`local.retention.bytes` 为日志段大小限制。
- `remote.storage.enable` 设置为 `true` 即可启用分层存储。

以下配置使用 local.retention.ms，但此属性可替换为 local.retention.bytes。此属性控制 Apache Kafka 将数据从主存储复制到分层存储之前可以经过的时长或 Apache Kafka 可以复制的字节数。有关支持的配置属性的更多详细信息，请参阅 Topic-level configuration。

> **Note**
>
> 您必须使用 Apache Kafka 客户端版本 3.0.0 及更高版本。这些版本仅在 `kafka-topics.sh` 的这些客户端版本中名为 `remote.storage.enable` 的设置。要对使用早期版本的 Apache Kafka 的现有主题启用分层存储，请参阅 在现有主题上启用分层存储 小节。

```
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
```

在现有主题上启用和禁用分层存储

这些小节介绍如何在已创建的主题上启用和禁用分层存储。要创建启用了分层存储的新集群和主题，请参阅使用 AWS Management Console创建启用了分层存储的集群。

在现有主题上启用分层存储

要在现有主题上启用分层存储，请使用以下示例中的 `alter` 命令语法。在已经存在的主题上启用分层存储后，您不会受到某个 Apache Kafka 客户端版本的限制。

```
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'
```

在现有主题上禁用分层存储

要在现有主题上禁用分层存储，请按照启用分层存储时的顺序使用 `alter` 命令语法。

```
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**
>
> 禁用分层存储后，就会完全删除分层存储中的主题数据。Apache Kafka 会保留主存储数据，但仍会应用基于 `local.retention.ms` 的主保留规则。禁用主题的分层存储后，便无法再次启用分层存储。要在现有主题上禁用分层存储，您不会受到某个 Apache Kafka 客户端版本的限制。

使用 AWS CLI 在现有集群上启用分层存储

> **Note**
>
> 只有在集群的 log.cleanup.policy 设置为 `delete` 时，才能启用分层存储，因为分层存储不支持压缩主题。如果未在该特定主题上启用分层存储，稍后可以将该主题的 log.cleanup.policy 配置为 `compact`。有关支持的配置属性的更多详细信息，请参阅 Topic-level configuration。

1. 更新 Kafka 版本：集群版本并非简单的整数。要查找集群的当前版本，请使用`DescribeCluster`操作或 `describe-cluster` AWS CLI 命令。示例版本是 `KTVPDKIKX0DER`。

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

2. 编辑集群存储模式。以下代码示例显示如何使用 update-storage API 将集群存储模式编辑为 TIERED。

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

使用控制台更新现有集群上的分层存储

> **Note**
>
> 只有在集群的 log.cleanup.policy 设置为 delete 时，才能启用分层存储，因为分层存储不支持压缩主题。如果未在该特定主题上启用分层存储，稍后可以将该主题的 log.cleanup.policy 配置为 compact。有关支持的配置属性的更多详细信息，请参阅 Topic-level configuration。

1. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
2. 转到集群摘要页面，再选择属性。
3. 转到存储部分，再选择编辑集群存储模式。
4. 先选择分层存储和 EBS 存储，再选择保存更改。

纵向扩展代理存储空间

您可以增加每个代理的 EBS 存储空间。您无法减少存储。

在此扩展操作期间，存储卷仍然可用。

> **Important**
>
> 扩展 MSK 集群的存储空间后，立即可以使用额外的存储空间。不过，在每次存储扩展事件之后，集群都需要一段冷却期。Amazon MSK 使用此冷却期来优化集群，再对其进行扩展。这段时间从最少 6 小时到超过 24 小时不等，具体取决于集群的存储大小和利用率以及流量。这既适用于 auto scaling 事件，也适用于使用该UpdateBrokerStorage操作进行手动缩放。有关正确调整存储空间大小的信息，请参阅 最佳实践。

您可以使用分层存储为代理纵向扩展到无限量的存储空间。请参阅分层存储。

主题

- 自动扩缩
- 手动扩展

自动扩缩

要自动扩展集群存储容量以应对使用量增加，您可以为 Amazon MSK 配置应用程序自动扩缩策略。在自动扩缩策略中，您可以设置目标磁盘利用率和最大扩展容量。

在为 Amazon MSK 使用自动扩缩之前，您应考虑以下几点：

- **Important**

  存储扩展操作每 6 小时只能发生一次。

  建议您从大小合适的存储卷开始，以满足存储需求。有关调整集群大小的指导，请参阅 调整集群的大小：每个集群的代理数量。
- Amazon MSK 不会因使用量减少而减小集群存储容量。Amazon MSK 不支持减小存储卷的大小。如果您需要减小集群存储大小，则必须将现有集群迁移到存储容量较小的集群。有关迁移集群的信息，请参阅 迁移。
- Amazon MSK 在亚太地区（大阪）和非洲（开普敦）区域不支持自动扩缩。
- 当您将自动扩缩策略与集群关联时，Amazon EC2 Auto Scaling 会自动创建用于目标跟踪的 Amazon CloudWatch 警报。如果您使用自动扩缩策略删除集群，则此 CloudWatch 警报仍然存在。要删除 CloudWatch 警报，您应该先从集群中删除自动扩缩策略，然后再删除集群。要了解有关目标跟踪的更多信息，请参阅《Amazon EC2 Auto Scaling用户指南》中的 Target tracking scaling policies for Amazon EC2 Auto Scaling。

自动扩缩策略详细信息

自动扩缩策略为集群定义以下参数：

- 存储利用率目标：Amazon MSK 用于触发自动扩缩操作的存储利用率阈值。您可以将此利用率目标设置为当前存储容量的 10% 到 80% 之间。建议您将“存储利用率目标”设置为 50% 到 60% 之间。

- 最大存储容量：Amazon MSK 可以为代理存储设置的最大扩展限值。您可以将每个代理的最大存储容量设置为最多 16TiB。有关更多信息，请参阅Amazon MSK 限额。

当 Amazon MSK 检测到 `Maximum Disk Utilization` 指标等于或大于 `Storage Utilization Target` 设置时，它会将存储容量增加 10GiB 或当前存储容量的 10%（以较大者为准）。例如，如果您有 1000GiB，则该数量为 100GiB。该服务每分钟检查一次存储利用率。进一步的扩展操作会继续增加存储容量，增幅为 10GiB 或当前存储容量的 10%（以较大者为准）。

要确定是否发生了自动扩缩操作，请使用 ListClusterOperations 操作。

为 Amazon MSK 集群设置自动扩缩

您可以使用 Amazon MSK 控制台、Amazon MSK API 或 AWS CloudFormation 为存储实现自动扩缩。CloudFormation 支持可通过 Application Auto Scaling 获得。

> **Note**
>
> 创建集群时，您无法实现自动扩缩。您必须先创建集群，然后为其创建并启用自动扩缩策略。但是，您可以在 Amazon MSK 服务创建集群时创建该策略。

使用 AWS Management Console 设置自动扩缩

1. 登录 AWS Management Console 并通过以下网址打开 Amazon MSK 控制台：https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。
2. 在集群列表中，选择集群。这会将您引导至列出集群详细信息的页面。
3. 在存储自动扩缩部分中，选择配置。
4. 创建并命名自动扩缩策略。指定存储利用率目标、最大存储容量和目标指标。
5. 选择 `Save changes`。

保存并启用新策略后，该策略将针对该集群变为活动状态。然后，当达到存储利用率目标时，Amazon MSK 会扩展集群的存储。

使用 CLI 设置自动扩缩

1. 使用 RegisterScalableTarget 命令注册存储利用率目标。
2. 使用 PutScalingPolicy 命令创建自动扩缩策略。

使用 API 设置自动扩缩

1. 使用 RegisterScalableTarget API 注册存储利用率目标。
2. 使用 PutScalingPolicy API 创建自动扩展策略。

手动扩展

要增加存储空间，请等待集群进入 ACTIVE 状态。存储扩展在两次事件之间至少有六个小时的冷却期。虽然该操作会立即提供更多存储空间，但该服务仍会需要 24 小时或更长时间对您的集群执行优化。这些优化会耗费的时长与存储的大小成正比。

使用扩展经纪商存储 AWS Management Console

1. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
2. 选择要更新代理存储的 MSK 集群。
3. 在存储部分中选择编辑。
4. 指定所需存储量。您只能增加存储量，不能减少存储量。
5. 选择 保存更改。

使用扩展经纪商存储 AWS CLI

运行以下命令，*ClusterArn*替换为您在创建集群时获得的 Amazon 资源名称 (ARN)。如果您没有该集群的 ARN，可以通过列出所有集群来找到它。有关更多信息，请参阅 the section called “列出集群”。

将 *Current-Cluster-Version* 替换为集群的当前版本。

> **Important**
>
> 集群版本不是简单的整数。要查找集群的当前版本，请使用DescribeCluster操作或 desc ribe-AWS CLI cluster 命令。示例版本是 KTVPDKIKX0DER。

Target-Volume-in-GiB 参数表示您希望每个代理具备的存储量。只能更新所有代理的存储。您不能指定要更新存储的单个代理。您为 *Target-Volume-in-GiB* 指定的值必须是大于 100 GiB 的整数。更新操作后每个代理的存储不能超过 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}'
```

使用 API 纵向扩展代理存储空间

要使用 API 更新代理存储，请参阅UpdateBrokerStorage。

预置存储吞吐量

Amazon MSK 代理会将数据保存在存储卷上。当生成器向集群写入数据、在代理之间复制数据以及使用器读取不在内存中的数据时，都会消耗存储 I/O。卷存储吞吐量是指是向存储卷写入数据和从存储卷读取数据的速率。预置存储吞吐量是指可为集群中的代理指定该速率的能力。

您可以为代理类型为 kafka.m5.4xlarge 或更大类型的且存储容量为 10GiB 或更高容量的集群，指定预置吞吐量速率（以每秒 MiB 为单位）。可以在创建集群期间指定预置吞吐量。您也可以为处于 ACTIVE 状态的集群启用或禁用预置吞吐量。

吞吐量瓶颈

代理吞吐量出现瓶颈的原因有很多：卷吞吐量、Amazon EC2 到 Amazon EBS 的网络吞吐量以及 Amazon EC2 的出口吞吐量。您可以启用预置存储吞吐量来调整卷吞吐量。不过，代理吞吐量限制可能是由 Amazon EC2 到 Amazon EBS 的网络吞吐量和 Amazon EC2 出口吞吐量造成。

Amazon EC2 出口吞吐量受使用器组数量和各使用器组使用器数量的影响。此外，对于较大的代理类型，Amazon EC2 到 Amazon EBS 网络吞吐量和 Amazon EC2 出口吞吐量都更高。

对于 10GiB 或更大的卷大小，您可以将存储吞吐量预置为每秒 250MiB 或更高值。默认为每秒 250MiB。要预配置存储吞吐量，必须选择代理类型 kafka.m5.4xlarge 或更大（或 kafka.m7g.2xlarge 或更大），并且可以指定最大吞吐量，如下表所示。

代理类型	最大存储吞吐量（MiB/s）
kafka.m5.4xlarge	593
kafka.m5.8xlarge	850
kafka.m5.12xlarge	1000
kafka.m5.16xlarge	1000

代理类型	最大存储吞吐量（MiB/s）
kafka.m5.24xlarge	1000
kafka.m7g.2xlarge	312.5
kafka.m7g.4xlarge	625
kafka.m7g.8xlarge	1000
kafka.m7g.12xlarge	1000
kafka.m7g.16xlarge	1000

测量存储吞吐量

您可以使用 VolumeReadBytes 和 VolumeWriteBytes 指标来衡量集群的平均存储吞吐量。使用这两个指标的总和得出以字节为单位的平均存储吞吐量。要获取集群的平均存储吞吐量，请将这两个指标设置为 SUM，将时长设置为 1 分钟，然后使用以下公式。

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

有关 VolumeReadBytes 和 VolumeWriteBytes 指标的信息，请参阅 the section called “PER_BROKER 级别监控”。

配置更新

您可以在开启预置吞吐量之前或之后更新 Amazon MSK 配置。不过，要想看到所需的吞吐量，您必须先执行这两个操作：更新 num.replica.fetchers 配置参数和开启预置吞吐量。

在默认 Amazon MSK 配置中，num.replica.fetchers 的值为 2。要更新 num.replica.fetchers，您可以使用下表中的建议值。这些值仅供参考。建议您根据自己的用例调整这些值。

代理类型	num.replica.fetchers
kafka.m5.4xlarge	4

代理类型	num.replica.fetchers
kafka.m5.8xlarge	8
kafka.m5.12xlarge	14
kafka.m5.16xlarge	16
kafka.m5.24xlarge	16

更新后的配置可能无法在 24 小时内生效，并且如果源卷未得到充分利用，则可能需要更长时间。不过，在迁移期间，过渡卷的性能至少等于源存储卷的性能。如果 1TiB 卷得到充分利用，通常约需六小时就能迁移到更新后的配置。

使用配置存储吞吐量 AWS Management Console

1. 登录并打开亚马逊 MSK 控制台，网址为 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。 AWS Management Console
2. 选择创建集群。
3. 选择自定义创建。
4. 指定集群的名称。
5. 在存储部分中选择启用。
6. 为各代理的存储吞吐量选择一个值。
7. 选择 VPC、可用区、子网和安全组。
8. 选择下一步。
9. 在安全步骤的底部，选择下一步。
10. 在监控和标记步骤的底部，选择下一步。
11. 检查集群设置，然后选择创建集群。

使用配置存储吞吐量 AWS CLI

本节举例说明如何使用创建启用了 AWS CLI 预配置吞吐量的集群。

1. 复制以下 JSON 并将其粘贴到文件中。将子网 ID 和安全组 ID 占位符替换为您账户的值。为文件 `cluster-creation.json` 命名并保存文件。

```
{
    "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. 从上一步中保存 JSON 文件的目录中运行以下 AWS CLI 命令。

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

使用 API 预置存储吞吐量

要在创建集群时配置预配置的存储吞吐量，请使用 CreateClusterV 2。

更新代理类型

您可以通过更改代理的类型（大小或系列）来按需扩展 MSK 集群，不必重新分配 Apache Kafka 分区。通过更改代理的类型，您可以根据工作负载的变化灵活地调整 MSK 集群的计算容量，不会中断集群 I/O。Amazon MSK 对指定集群中的所有代理使用相同的代理类型。本节旨在介绍如何更新 MSK 集群的代理类型。当集群启动并运行后，将以滚动方式更新代理类型。这意味着 Amazon MSK 一次关闭一个代理来更新代理类型。有关如何在代理类型更新期间使集群高度可用的信息，请参阅 the section called “构建高度可用的集群”。为了进一步降低对生产力的任何潜在影响，您可以在流量较低的时期更新代理类型。

代理类型更新期间，您可以继续生成和使用数据。不过，您必须等到更新完成，才能重启代理或调用 Amazon MSK operations 下列出的任何更新操作。

如果想将集群更新为较小的代理类型，建议您先在测试集群上尝试更新，了解会对场景产生的影响。

> **Important**
>
> 如果每个代理的分区数超过 the section called “ 调整集群的大小：每个代理的分区数量” 中指定的最大数量，则无法将集群更新为较小的代理类型。

使用更新经纪商类型 AWS Management Console

1. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
2. 选择要更新代理类型的 MSK 集群。
3. 在集群的详细信息页面上，找到代理摘要部分，从中选择编辑代理类型。
4. 从列表中选择所需的代理类型。
5. 保存更改。

使用更新经纪商类型 AWS CLI

1. 运行以下命令，*ClusterArn*替换为您在创建集群时获得的 Amazon 资源名称 (ARN)。如果您没有该集群的 ARN，可以通过列出所有集群来找到它。有关更多信息，请参阅 the section called “列出集群”。

   将*######*替换为集群的当前版本*TargetType*以及您希望代理成为的新类型。要了解有关代理类型的更多信息，请参阅 the section called “代理类型”。

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

下面的示例说明如何使用此命令：

```
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
```

该 命令的输出如以下 JSON 示例所示。

```
{
    "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. 要获得update-broker-type操作结果，请运行以下命令，*ClusterOperationArn*替换为在命令输出中获得的 ARN。update-broker-type

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

该 describe-cluster-operation 命令的输出如以下 JSON 示例所示。

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

如果 OperationState 的值为 UPDATE_IN_PROGRESS，请等待一段时间，然后再次运行 describe-cluster-operation 命令。

使用 API 更新代理类型

要使用 API 更新代理类型，请参阅UpdateBrokerType。

更新 Amazon MSK 集群的配置

要更新集群配置，请确保集群处于 ACTIVE 状态。您还必须确保 MSK 集群上每个代理的分区数低于 the section called “ 调整集群的大小：每个代理的分区数量” 中所述的限制。如果超过这些限制，便无法更新集群的配置。

有关 MSK 配置的信息，包括如何创建自定义配置、可以更新哪些属性以及更新现有集群的配置时会发生什么情况，请参阅 配置。

使用更新集群的配置 AWS CLI

1. 复制以下 JSON 并将其保存到文件中。将文件命名为 configuration-info.json。*ConfigurationArn*替换为您要用于更新集群的配置的 Amazon 资源名称 (ARN)。在以下 JSON 中，ARN 字符串必须使用引号引起来。

   将 *Configuration-Revision* 替换为要使用的配置的修订版本。配置修订版本是从 1 开始的整数。在以下 JSON 中，该整数不能使用引号引起来。

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

2. 运行以下命令，*ClusterArn*替换为创建集群时获得的 ARN。如果您没有该集群的 ARN，可以通过列出所有集群来找到它。有关更多信息，请参阅 the section called “列出集群”。

将 *Path-to-Config-Info-File* 替换为您的配置信息文件的路径。如果您将上一步中创建的文件命名为 configuration-info.json，并将其保存在当前目录中，*Path-to-Config-Info-File* 即为 configuration-info.json。

将 *Current-Cluster-Version* 替换为集群的当前版本。

> **Important**
>
> 集群版本不是简单的整数。要查找集群的当前版本，请使用DescribeCluster操作或 describe- AWS CLI cluster 命令。示例版本是 KTVPDKIKX0DER。

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

下面的示例说明如何使用此命令：

```
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"
```

该 update-cluster-configuration 命令的输出如以下 JSON 示例所示。

```
{
    "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. 要获得update-cluster-configuration操作结果，请运行以下命令，*ClusterOperationArn*替换为在命令输出中获得的 ARN。update-cluster-configuration

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

该 describe-cluster-operation 命令的输出如以下 JSON 示例所示。

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

在此输出中，OperationType 是 UPDATE_CLUSTER_CONFIGURATION。如果 OperationState 的值为 UPDATE_IN_PROGRESS，请等待一段时间，然后再次运行 describe-cluster-operation 命令。

使用 API 更新集群的配置

要使用 API 更新集群的配置，请参阅UpdateClusterConfiguration。

扩展 Amazon MSK 集群

如果要增加 MSK 集群中的代理数量，请使用此 Amazon MSK 操作。要扩展集群，请确保集群处于 ACTIVE 状态。

> **Important**
>
> 如果要扩展 MSK 集群，请确保使用此 Amazon MSK 操作。切勿尝试在未使用此操作的情况下向集群添加代理。

有关在将代理添加到集群后如何重新平衡分区的信息，请参阅the section called “重新分配分区”。

使用扩展集群 AWS Management Console

1. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
2. 选择要增加代理数量的 MSK 集群。
3. 在集群详细信息页面上，选择集群级代理详细信息标题旁边的编辑按钮。
4. 输入您希望集群在每个可用区具有的代理数量，然后选择保存更改。

使用扩展集群 AWS CLI

1. 运行以下命令，*ClusterArn*替换为您在创建集群时获得的 Amazon 资源名称 (ARN)。如果您没有该集群的 ARN，可以通过列出所有集群来找到它。有关更多信息，请参阅 the section called “列出集群”。

   将 *Current-Cluster-Version* 替换为集群的当前版本。

   > **Important**
   >
   > 集群版本不是简单的整数。要查找集群的当前版本，请使用DescribeCluster操作或 describe- AWS CLI cluster 命令。示例版本是 KTVPDKIKX0DER。

   Target-Number-of-Brokers 参数表示在此操作成功完成时您希望集群具有的代理节点的总数。您为 *Target-Number-of-Brokers* 指定的值必须是大于集群中当前代理数量的整数。它还必须是可用区数目的倍数。

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

   该 update-broker-count 操作的输出如以下 JSON 所示：

```
{

    "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. 要获得update-broker-count操作结果，请运行以下命令，*ClusterOperationArn*替换为在命令输出中获得的 ARN。update-broker-count

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

该 describe-cluster-operation 命令的输出如以下 JSON 示例所示。

```
{
    "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
        }
    }
}
```

在此输出中， OperationType 是 INCREASE_BROKER_COUNT。如果 OperationState 的值为 UPDATE_IN_PROGRESS，请等待一段时间，然后再次运行 describe-cluster-operation 命令。

使用 API 扩展集群

要使用 API 增加集群中代理的数量，请参阅UpdateBrokerCount。

更新集群的安全设置

使用此 Amazon MSK 操作更新 MSK 集群的身份验证设置和客户端到代理加密设置。您还可以更新用于签署证书以进行双向 TLS 身份验证的私有安全证书颁发机构。您无法更改集群内 (broker-to-broker) 加密设置。

集群必须处于 ACTIVE 状态才能更新安全设置。

如果启用使用 IAM、SASL 或 TLS 的身份验证，您还必须开启客户端和代理之间的加密。下表显示了可能的组合。

身份验证	客户端到代理加密选项	代理到代理加密
无身份验证	TLS、PLAINTEXT、TLS_PLAINTEXT	可以开启或关闭。
mTLS	TLS、TLS_PLAINTEX	必须打开。
SASL/SCRAM	TLS	必须打开。
SASL/IAM	TLS	必须打开。

当客户端到代理加密设置为 TLS_PLAINTEXT，且客户端到身份验证设置为 mTLS 时，Amazon MSK 会创建两种类型的侦听器供客户端连接：一种是供客户端在使用 mTLS 身份验证和 TLS 加密的情况下进行连接，另一种是供客户端在不使用身份验证或加密的情况下进行连接（明文）。

有关安全设置的更多信息，请参阅 安全性。

使用更新集群的安全设置 AWS Management Console

1. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
2. 选择要更新的 MSK 集群。
3. 在设置部分中选择编辑。
4. 选择集群所需的身份验证和加密设置，然后选择保存更改。

使用更新集群的安全设置 AWS CLI

1. 创建一个 JSON 文件，内含您希望集群具有的加密设置。以下是示例。

   > **Note**
   >
   > 您只能更新客户端到代理加密设置。您无法更新集群内 (broker-to-broker) 加密设置。

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

2. 创建一个 JSON 文件，内含您希望集群具有的身份验证设置。以下是示例。

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

3. 运行以下 AWS CLI 命令：

   ```
   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
   ```

   该 `update-security` 操作的输出如以下 JSON 所示：

   ```
   {

       "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. 要查看`update-security`操作的状态，请运行以下命令，*ClusterOperationArn*替换为在命令输出中获得的 ARN。`update-security`

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

   该 `describe-cluster-operation` 命令的输出如以下 JSON 示例所示。

   ```
   {
   ```

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

如果 `OperationState` 的值为 `PENDING` 或 `UPDATE_IN_PROGRESS`，请等待一段时间，然后再次运行 `describe-cluster-operation` 命令。

使用 API 更新集群的安全设置

要使用 API 更新集群的安全设置，请参阅UpdateSecurity。

> **Note**
>
> 用于更新集群安全设置的 AWS CLI 和 API 操作是等效的。这意味着，如果您调用安全更新操作并指定与集群当前设置相同的身份验证或加密设置，则该设置不会更改。

重启 Amazon MSK 集群的代理

如果要重启 MSK 集群的代理，请使用此 Amazon MSK 操作。要重启集群的代理，请确保集群处于 `ACTIVE` 状态。

在系统维护（例如修补或版本升级）期间，Amazon MSK 服务可能会重启 MSK 集群的代理。您可以通过手动重启代理来测试 Kafka 客户端的弹性，据此确定客户端对系统维护的响应情况。

使用 AWS Management Console 重启代理

1. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
2. 选择要重启代理的 MSK 集群。

3. 向下滚动到代理详细信息部分，然后选择要重启的代理。
4. 选择重启代理按钮。

使用 AWS CLI 重启代理

1. 运行以下命令，将 *ClusterArn* 替换为创建集群时所获取的Amazon 资源名称（ARN），将 *BrokerId* 替换为要重启的代理的 ID。

   > **Note**
   >
   > reboot-broker 操作一次只支持重启一个代理。

   如果您没有该集群的 ARN，可以通过列出所有集群来找到它。有关更多信息，请参阅the section called “列出集群”。

   如果没有集群的代理 ID，您可以通过列出代理节点来找到相应 ID。有关更多信息，请参阅 list-nodes。

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

   该 reboot-broker 操作的输出如以下 JSON 所示：

   ```
   {

       "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. 要获取 reboot-broker 操作的结果，请运行以下命令，将 *ClusterOperationArn* 替换为您在 reboot-broker 命令的输出中获得的 ARN。

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

   该 describe-cluster-operation 命令的输出如以下 JSON 示例所示。

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

重新启动操作完成后，`OperationState` 处于 `REBOOT_COMPLETE` 状态。

使用 API 重启代理

要使用 API 重启集群中的代理，请参阅 RebootBroker。

在修补和其他维护期间代理重启的影响

Amazon MSK 会定期更新您的经纪商的软件。如果您遵循最佳实践，这些更新不会影响应用程序的写入和读取。

Amazon MSK 使用软件的滚动更新来保持集群的高可用性。在此过程中，经纪人会逐一重启，Kafka 会自动将领导权移交给另一家在线经纪商。Kafka 客户端具有内置机制，可以自动检测分区领导层的变化，并继续向 MSK 集群中写入和读取数据。

在经纪商下线后，您的客户端上会出现暂时断开连接错误是正常的。您还将观察到写入和读取延迟出现短暂的峰值（通常为几分钟）。这些峰值是预料之中的，是由客户重新连接到新的领导者经纪商造成的。

您还将观察到该指标的增加`UnderReplicatedPartitions`，这是预期的，因为已关闭的代理上的分区不再复制数据。这不会影响应用程序的写入和读取，因为托管在其他代理上的这些分区的副本现在正在处理请求。

软件更新后，当经纪人重新上线时，它需要 “catch” 离线时生成的消息。在 catch up 期间，您可能还会观察到卷吞吐量和 CPU 的使用率有所增加。如果您的代理上有足够的 CPU、内存、网络和卷资源，则这些不会对集群的写入和读取产生任何影响。

为 Amazon MSK 集群添加标签

您可以将自己的元数据以标签的形式分配给 Amazon MSK 资源，例如 MSK 集群。标签是您为资源定义的键值对。使用标签是管理 AWS 资源和组织数据（包括账单数据）的一种简单却强有力的方式。

主题

有关标签的基本知识

可使用 Amazon MSK API 完成以下任务：

- 将标签添加到 Amazon MSK 资源。
- 列出 Amazon MSK 资源的标签。
- 从 Amazon MSK 资源中删除标签。

您可以使用标签对 Amazon MSK 资源进行分类。例如，您可以按用途、所有者或环境对 Amazon MSK 集群进行分类。由于您定义每个标签的键和值，因此您可以创建一组自定义类别来满足您的特定需求。例如，您可以定义一组标签来帮助您按所有者和关联应用程序跟踪集群。

以下是标签的多个示例：

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

使用标签跟踪成本

您可以使用标签对 AWS 成本进行分类和跟踪。当您将标签应用于 AWS 资源（包括 Amazon MSK 集群）时，您的 AWS 成本分配报告将包括按标签汇总的使用率和成本。您可通过应用代表业务类别（如成本中心、应用程序名称或拥有者）的标签来整理多种服务的成本。有关更多信息，请参阅 AWS Billing 用户指南中的对自定义账单报告使用成本分配标签。

标签限制

以下限制适用于 Amazon MSK 中的标签。

基本限制

- 每个资源的最大标签数是 50。
- 标签键和值区分大小写。
- 无法更改或编辑已删除的资源的标签。

标签键限制

- 每个标签键必须是唯一的。如果您添加的标签具有已使用的键，则您的新标签将覆盖现有键值对。
- 标签键不能以 `aws:` 开头，因为此前缀将预留以供 AWS 使用。AWS 将代表您创建以此前缀开头的标签，但您不能编辑或删除这些标签。
- 标签键的长度必须介于 1 和 128 个 Unicode 字符之间。
- 标签键必须包含以下字符：Unicode 字母、数字、空格和以下特殊字符：`_ . / = + - @`。

标签值限制

- 标签值的长度必须介于 0 和 255 个 Unicode 字符之间。
- 标签值可以为空。另外，它们必须包含以下字符：Unicode 字母、数字、空格和以下任意特殊字符：`_ . / = + - @`。

使用 Amazon MSK API 为资源添加标签

您可以使用以下操作来为 Amazon MSK 资源添加标签或取消添加标签，或者列出资源的当前标签集：

- ListTagsForResource

- TagResource
- UntagResource

通过 Amazon MSK 控制台访问 Amazon EventBridge Pipes

Amazon EventBridge Pipes 可将来源与目标连接起来。管道用于支持的来源和目标之间的点对点集成，并支持高级转换和扩充。EventBridge Pipes 提供了一种高度可扩展的方式，可以将 Amazon MSK 集群连接到 Step Functions、Amazon SQS 和 API Gateway 等 AWS 服务，以及 Salesforce 等第三方软件即服务（SaaS）应用程序。

要设置管道，请选择来源、添加可选筛选、定义可选扩充，然后为事件数据选择目标。

在 Amazon MSK 集群的详细信息页面上，您可以查看使用该集群作为来源的管道。您还可以在此页面上：

- 启动 EventBridge 控制台来查看管道详细信息。
- 启动 EventBridge 控制台来创建一个以集群为来源的新管道。

有关将 Amazon MSK 集群配置为管道来源的更多信息，请参阅《Amazon EventBridge User Guide》中的 Amazon Managed Streaming for Apache Kafka cluster as a source。有关 EventBridge Pipes 的一般详细信息，请参阅 EventBridge Pipes。

访问指定 Amazon MSK 集群的 EventBridge 管道

1. 打开 Amazon ECS 控制台并选择集群。
2. 选择一个集群。
3. 在集群详细信息页面上，选择集成选项卡。

   集成选项卡包含当前配置为使用所选集群作为来源的所有管道的列表，包括：

   - 管道名称
   - 当前状态
   - 管道目标
   - 上次修改管道的时间
4. 根据需要管理 Amazon MSK 集群的管道：

   访问有关管道的更多详细信息

- 选择管道。

  此操作会启动 EventBridge 控制台的管道详细信息页面。

创建新管道

- 选择将 Amazon MSK 集群连接到管道。

  此操作会启动 EventBridge 控制台的创建管道页面，并将 Amazon MSK 集群指定为管道来源。有关更多信息，请参阅《Amazon EventBridge User Guide》中的 Creating an EventBridge pipe。

- 您也可以从集群页面为集群创建管道。选择集群，然后从操作菜单中选择创建 EventBridge 管道。

Amazon MSK 配置

适用于 Apache 的亚马逊托管流媒体 Kafka 为代理、主题和 Apache 节点提供了默认配置。ZooKeeper 您还可以创建自定义配置，并使用这些配置来创建新的 MSK 集群或更新现有集群。MSK 配置由一组属性及其相应的值构成。

主题

自定义 MSK 配置

您可以使用 Amazon MSK 来创建自定义 MSK 配置，并可以在该配置中设置以下属性。未显式设置的属性将获得其在 the section called “默认配置” 中具有的值。有关配置属性的更多信息，请参阅 Apache Kafka 配置。

Apache Kafka 配置属性

名称	描述
allow.everyone.if.no.acl.found	如果要将此属性设置为 `false`，请务必先为集群定义 Apache Kafka ACL。如果将此属性设置为 `false`，且没有先定义 Apache Kafka ACL，则将失去对集群的访问权限。如果发生这种情况，您可以再次更新配置并将此属性设置为 `true`，以重新获得对集群的访问权限。
auto.create.topics.enable	在服务器上启用主题自动创建。
compression.type	给定主题的最终压缩类型。可以将此属性设置为标准压缩编解码器（`gzip`、`snappy`、`lz4` 和 `zstd`）。它还接受 `uncompressed` 。此值等同于不压缩。如果将该值设置为 `producer`，则意味着保留生成器设置的原始压缩编解码器。

名称	描述
connections.max.idle.ms	空闲连接超时（以毫秒为单位）。如果连接的空闲时间超过您为此属性设置的值，服务器套接字处理器线程会关闭这些连接。
default.replication.factor	自动创建的主题的默认复制因子。
delete.topic.enable	启用删除主题操作。如果禁用此设置，则无法通过管理工具删除主题。
group.initial.rebalance.delay.ms	在组协调器执行第一次重新平衡之前，组协调器等待更多数据使用器加入新组的时间。更长的延迟时间意味着重新平衡可能会更少，但这会增加处理开始之前的时间。
group.max.session.timeout.ms	注册使用器的最长会话超时时间。超时时间越长，可供使用器用来处理检测信号之间的消息的时间就越多，但这会导致需要花更多时间来检测故障。
group.min.session.timeout.ms	注册使用器的最短会话超时时间。超时时间越短，故障检测的速度就会越快，但需要更频繁的使用器检测信号。这可能会使代理资源不堪重负。
leader.imbalance.per.broker.percentage	各代理允许的领导节点不平衡比率。如果各代理超过了此值，则控制器将触发领导平衡操作。此值以百分比的形式指定。
log.cleaner.delete.retention.ms	您希望 Apache Kafka 保留已删除的记录的时间量。最小值为 0。

名称	描述
log.cleaner.min.cleanable.ratio	此配置属性的值可介于 0 到 1 之间。此值决定日志压缩器尝试清理日志的频率（如果日志压缩已启用）。默认情况下，如果已压缩超过 50% 的日志，Apache Kafka 会避免清理日志。这一比率限制了日志因重复项浪费的最大空间（该值为 50%，这意味着最多有 50% 的日志可能是重复的）。更高的比率意味着更少、更高效的清理，但也意味着会浪费更多的日志空间。
log.cleanup.policy	超出保留时段的分段的默认清除策略。有效策略的逗号分隔列表。有效策略为 `delete` 和 `compact`。对于启用了分层存储的集群，有效策略为仅 `delete`。
log.flush.interval.messages	将消息刷新到磁盘之前，日志分区上累积的消息的数量。
log.flush.interval.ms	任何主题中的消息在刷新到磁盘之前保存在内存中的最长时间（以毫秒为单位）。如果未设置此值，则使用 log.flush.scheduler.interval.ms 中的值。最小值为 0。
log.message.timestamp.difference.max.ms	代理收到消息时的时间戳与消息中指定的时间戳之间的最大时间差。如果 log.message.timestamp.type=CreateTime，则如果时间戳的差异超过此阈值，则消息将被拒绝。如果 log.message.timestamp.type LogAppendTime =，则忽略此配置。
log.message.timestamp.type	指定消息中的时间戳是消息创建时间还是日志追加时间。允许的值是 `CreateTime` 和 `LogAppendTime` 。
log.retention.bytes	删除日志前的最大日志大小。

名称	描述
log.retention.hours	删除日志文件前保留日志文件的小时数，它是 log.retention.ms 属性的三级属性。
log.retention.minutes	删除日志文件前保留日志文件的分钟数，它是 log.retention.ms 属性的二级属性。如果未设置此值，则使用 log.retention.hours 中的值。
log.retention.ms	删除日志文件前保留日志文件的毫秒数（以毫秒为单位）。如果未设置，则使用 log.retention.minutes 中的值。
log.roll.ms	推出新日志段之前的最长时间（以毫秒为单位）。如果未设置此属性，则使用 log.roll.hours 中的值。此属性的最小可能值为 1。
log.segment.bytes	单个日志文件的最大大小。
max.incremental.fetch.session.cache.slots	维护的增量提取会话的最大数量。
message.max.bytes	Kafka 允许的最大记录批处理大小。如果增加此值，并且存在大于 0.10.2 的使用器，则使用器的提取大小也必须增加，以便它们能够提取如此大的记录批处理。 最新的消息格式版本总是将消息分组到批处理中来提高效率。以前的消息格式版本不会将未压缩的记录分组到批处理中，在此情况下，此限制仅适用于单条记录。 可使用主题级别 max.message.bytes 配置为每个主题设置此值。

名称	描述
min.insync.replicas	当生成器将 acks 设置为 "all"（或 "-1"）时，min.insync.replicas 中的值会指定为使写入被视为成功而必须确认写入的最小副本数。如果无法达到此最低限度，则生产者会引发异常（ NotEnoughReplicas 或 NotEnoughReplicasAfterAppend ）。 您可以使用 min.insync.replicas 和 acks 中的值来强制执行更大的持久性保证。例如，您可以创建复制因子为 3 的主题，将 min.insync.replicas 设置为 2，并在 acks 为 "all" 的情况下进行生成。这可确保在大多数副本未收到写操作时，创建器将引发异常。
num.io.threads	服务器用于处理请求的线程的数目，其中可能包括磁盘 I/O。
num.network.threads	服务器用于接收来自网络的请求并向其发送响应的线程数量。
num.partitions	每个主题的默认日志分区数。
num.recovery.threads.per.data.dir	在启动时用于日志恢复以及在关闭时用于刷新的每个数据目录的线程数量。
num.replica.fetchers	用于从源代理复制消息的提取器线程数。增大此值会增加跟踪器代理中的 I/O 并行度。
offsets.retention.minutes	当一个使用器组丢失其所有使用器（即变空）后，其偏移量将在此保留期内保留，然后被丢弃。对于独立使用器（即，使用手动分配的使用器），偏移量会在最后一次提交时间加上此保留期后过期。
offsets.topic.replication.factor	偏移量主题的复制因子。将此值设置得更高可以确保可用性。内部主题创建失败，直到集群大小满足此复制因子要求。

名称	描述
replica.fetch.max.bytes	尝试为每个分区提取的消息的字节数。这不是绝对最大值。如果提取的第一个非空分区中的第一个记录批处理大于此值，则将返回该记录批处理以确保取得进展。message.max.bytes（代理配置）或 max.message.bytes（主题配置）定义代理接受的最大记录批处理大小。
replica.fetch.response.max.bytes	整个提取响应预期的最大字节数。记录是分批提取的，如果提取的第一个非空分区中的第一个记录批处理大于此值，则仍将返回该记录批处理以确保取得进展。这不是绝对最大值。message.max.bytes（代理配置）或 max.message.bytes（主题配置）属性指定代理接受的最大记录批处理大小。
replica.lag.time.max.ms	如果跟踪器没有发送任何提取请求，或者至少在此毫秒数内没有使用到领导的日志结束偏移量，则领导会从 ISR 中删除追随者。 MinValue: 10000 MaxValue = 30000
replica.selector.class	实现 ReplicaSelector的完全限定类名。代理使用此值来查找首选读取副本。如果您使用的是 Apache Kafka 版本 2.4.1 或更高版本，并且希望允许使用器从最近的副本提取，请将此属性设置为 `org.apache.kafka.common.replica.RackAwareReplicaSelector` 。有关更多信息，请参阅 the section called “Apache Kafka 版本 2.4.1（改用 2.4.1.1 版）”。
replica.socket.receive.buffer.bytes	网络请求的套接字接收缓冲区。

名称	描述
socket.receive.buffer.bytes	套接字服务器套接字的 SO_RCVBUF 缓冲区。可为此属性设置的最小值为 -1。如果该值为 -1，则 Amazon MSK 使用 OS 默认值。
socket.request.max.bytes	套接字请求中的最大字节数。
socket.send.buffer.bytes	套接字服务器套接字的 SO_SNDBUF 缓冲区。可为此属性设置的最小值为 -1。如果该值为 -1，则 Amazon MSK 使用 OS 默认值。
transaction.max.timeout.ms	事务的最大超时时间。如果客户请求的交易时间超过此值，则经纪商会在中返回错误 InitProducerIdRequest。这可防止客户端的超时时间过长，且此情况可能会导致使用器无法阅读事务中包含的主题。
transaction.state.log.min.isr	事务主题的已覆盖 min.insync.replicas 配置。
transaction.state.log.replication.factor	事务主题的复制因子。将此属性设置为较高的值可提高可用性。内部主题创建失败，直到集群大小满足此复制因子要求。
transactional.id.expiration.ms	事务协调器在其事务 ID 过期之前，等待接收当前事务的任何事务状态更新的时间（以毫秒为单位）。此设置还会影响生成器 ID 的到期时间，因为它会导致生成器 ID 在最后一次使用给定生成器 ID 写入之后过期。如果由于主题的保留设置而删除了生成器 ID 的最后一次写入内容，则生成器 ID 可能会提前过期。此属性的最小值为 1 毫秒。
unclean.leader.election.enable	表示不在 ISR 集中的副本是否应作为最后手段充当领导，即使这可能会导致数据丢失。

名称	描述
zookeeper.connection.timeout.ms	客户端等待与之建立连接的最长时间。ZooKeeper如果未设置此值，则使用 zookeeper.session.timeout.ms 中的值。 MinValue = 6000 MaxValue （含）= 18000
zookeeper.session.timeout.ms	Apache ZooKeeper 会话超时时间（以毫秒为单位）。 MinValue = 6000 MaxValue （含）= 18000

要了解如何创建自定义 MSK 配置、列出所有配置或描述它们，请参阅 the section called “配置操作”。要使用自定义 MSK 配置创建 MSK 集群或使用新的自定义配置更新集群，请参阅 工作方式。

当您使用自定义 MSK 配置更新现有 MSK 集群时，Amazon MSK 会在必要时重新开始滚动，并使用最佳实践来最大程度地减少客户停机时间。例如，在 Amazon MSK 重新启动每个代理后，Amazon MSK 会尝试让代理获得其在配置更新期间可能缺失的数据，然后再移至下一个代理。

动态配置

除了 Amazon MSK 提供的配置属性之外，您还可以动态设置不要求代理重新启动的集群级别和代理级别的配置属性。您可以动态设置一些配置属性。这些是 Apache Kafka 文档中代理配置下的表中未标记为只读的属性。有关动态配置和示例命令的信息，请参阅 Apache Kafka 文档中的更新代理配置。

> **Note**
>
> 您可以设置 `advertised.listeners` 属性，但不能设置 `listeners` 属性。

主题级别的配置

您可以使用 Apache Kafka 命令为新主题和现有主题设置或修改主题级别的配置属性。有关主题级别的配置属性以及如何设置这些属性之示例的更多信息，请参阅 Apache Kafka 文档中的 Topic-Level Configs。

配置状态

Amazon MSK 配置可以处于以下某种状态。要对配置执行操作，该配置必须处于 ACTIVE 或 DELETE_FAILED 状态：

- ACTIVE
- DELETING
- DELETE_FAILED

Amazon MSK 的默认配置

在未指定自定义 MSK 配置的情况下创建 MSK 集群时，Amazon MSK 会创建默认配置，并将此配置与下表中显示的值结合使用。对于不在此表中的属性，Amazon MSK 将使用与您的 Apache Kafka 版本关联的默认值。有关这些默认值的列表，请参阅 Apache Kafka 配置。

默认配置值

名称	描述	非分层存储集群的默认值	启用了分层存储的集群的默认值
allow.everyone.if.no.acl.found	如果没有与特定资源匹配的资源模式，则该资源没有关联的 ACL。在本例中，如果将此属性设置为 true，则所有用户（而不仅仅是超级用户）均可访问该资源。	true	true
auto.create.topics.enable	在服务器上启用主题的自动创建。	false	false

名称	描述	非分层存储集群的默认值	启用了分层存储的集群的默认值
auto.leader.rebalance.enable	启用自动领导平衡。如果需要，后台线程会定期检查并启动领导平衡。	`true`	`true`
default.replication.factor	自动创建的主题的默认复制因子。	3 表示位于 3 个可用区中的集群，2 表示位于 2 个可用区中的集群。	3 表示位于 3 个可用区中的集群，2 表示位于 2 个可用区中的集群。
local.retention.bytes	分区在删除旧日志段之前的最大本地日志段大小。如果未设置此值，则使用 log.retention.bytes 中的值。有效值应始终小于或等于 log.retention.bytes 值。默认值 -2 表示对本地保留没有限制。这与 retention.ms/bytes 的设置 -1 相对应。local.retention.ms 和 local.retention.bytes 属性与 log.retention 类似，因为它们用于确定日志段应在本地存储中保留多长时间。现有的 log.retention.* 配置是主题分区的保留配置。这包括本地存储和远程存储。有效值：[-2; +Inf] 中的整数	-2 表示无限制	-2 表示无限制

名称	描述	非分层存储集群的默认值	启用了分层存储的集群的默认值
local.retention.ms	删除之前保留本地日志段的毫秒数。如果未设置此值，Amazon MSK 使用 log.retention.ms 中的值。有效值应始终小于或等于 log.retention.bytes 值。默认值 -2 表示对本地保留没有限制。这与 retention.ms/bytes 的设置 -1 相对应。 local.retention.ms 和 local.retention.bytes 值类似于 log.retention。MSK 使用此配置确定日志段应在本地存储中保留多长时间。现有的 log.retention.* 配置是主题分区的保留配置。这包括本地存储和远程存储。有效值为大于 0 的整数。	-2 表示无限制	-2 表示无限制

名称	描述	非分层存储集群的默认值	启用了分层存储的集群的默认值
log.message.timestamp.difference.max.ms	代理收到消息时的时间戳与消息中指定的时间戳之间允许的最大差异。如果 log.message.timestamp.type=CreateTime，则如果时间戳的差异超过此阈值，则消息将被拒绝。如果 log.message.timestamp.type LogAppendTime =，则忽略此配置。允许的最大时间戳差异不应大于 log.retention.ms，以避免不必要的频繁日志滚动。	9223372036854775807	Kafka 2.8.2. 分层 86400000
log.segment.bytes	单个日志文件的最大大小。	1073741824	134217728

名称	描述	非分层存储集群的默认值	启用了分层存储的集群的默认值
min.insync.replicas	当生成器将 acks 的值（确认生成器从 Kafka 代理获得）设置为 "all"（或 "-1"）时，min.insync.replicas 中的值会指定为使写入被视为成功而必须确认写入的最小副本数。如果此值未达到此最小值，则生产者会引发异常（ NotEnoughReplicas 或 NotEnoughReplicasAfterAppend）。 当您同时使用 min.insync.replicas 和 acks 中的值时，可以强制执行更大的持久性保证。例如，您可以创建复制因子为 3 的主题，将 min.insync.replicas 设置为 2，并在 acks 为 "all" 的情况下进行生成。这可确保在大多数副本未收到写操作时，创建器将引发异常。	2 表示位于 3 个可用区中的集群，1 表示位于 2 个可用区中的集群。	2 表示位于 3 个可用区中的集群，1 表示位于 2 个可用区中的集群。
num.io.threads	服务器用于生成请求的线程的数量，其中可能包括磁盘 I/O。	8	max(8, vCPUs)，其中 vCPU 取决于代理的实例大小

名称	描述	非分层存储集群的默认值	启用了分层存储的集群的默认值
num.network.threads	服务器用于接收来自网络的请求并向网络发送响应的线程数量。	5	max(5, vCPUs / 2)，其中 vCPU 取决于代理的实例大小
num.partitions	每个主题的默认日志分区数。	1	1
num.replica.fetchers	用于从源代理复制消息的提取器线程数量。如果增加此值，则可以增加跟踪器代理中的 I/O 并行度。	2	max(2, vCPUs / 4)，其中 vCPU 取决于代理的实例大小
remote.log.msk.disable.policy	与 remote.storage.enable 一起使用以禁用分层存储。将此策略设置为“删除”，以表示在将 remote.storage.enable 设置为 false 时，分层存储中的数据将被删除。	不适用	删除
remote.log.reader.threads	远程日志读取器线程池大小，用于安排任务以从远程存储中提取数据。	不适用	max(10, vCPUs * 0.67)，其中 vCPU 取决于代理的实例大小

名称	描述	非分层存储集群的默认值	启用了分层存储的集群的默认值
remote.storage.enable	如果设置为 true，则为主题启用分层（远程）存储。如果设置为 false，且 remote.log.msk.disable.policy 设置为“删除”，则禁用主题级别的分层存储。禁用分层存储时，会从远程存储中删除数据。禁用主题的分层存储时，无法再次启用分层存储。	false	true
replica.lag.time.max.ms	如果跟踪器没有发送任何提取请求，或者至少在此毫秒数内没有使用到领导的日志结束偏移量，则领导会从 ISR 中删除追随者。	30000	30000

名称	描述	非分层存储集群的默认值	启用了分层存储的集群的默认值
retention.ms	必填字段。最短时间为 3 天。该设置是强制性的，因此没有默认值。 Amazon MSK 使用 retention.ms value 与 local.retention.ms 来确定数据何时从本地存储移动到分层存储。local.retention.ms 值指定何时将数据从本地存储移动到分层存储。retention.ms 值指定何时从分层存储中删除数据（即从集群中删除）。有效值：[-1; +Inf] 中的整数	最少 259,200,000 毫秒（3 天）。-1 表示无限保留。	最少 259,200,000 毫秒（3 天）。-1 表示无限保留。
socket.receive.buffer.bytes	套接字服务器套接字的 SO_RCVBUF 缓冲区。如果值为 -1，则使用操作系统默认值。	102400	102400
socket.request.max.bytes	套接字请求中的最大字节数。	104857600	104857600
socket.send.buffer.bytes	套接字服务器套接字的 SO_SNDBUF 缓冲区。如果值为 -1，则使用操作系统默认值。	102400	102400

名称	描述	非分层存储集群的默认值	启用了分层存储的集群的默认值
unclean.leader.election.enable	表示您是否希望不在 ISR 集中的副本作为最后手段充当领导，即使这可能会导致数据丢失。	true	false
zookeeper.session.timeout.ms	Apache ZooKeeper 会话超时时间（以毫秒为单位）。	18000	18000
zookeeper.set.acl	将客户端设置为使用安全 ACL。	false	false

有关如何指定自定义配置值的信息，请参阅 the section called “自定义 配置”。

分层存储主题级别的配置指南

以下是在主题级别配置分层存储时的默认设置和限制。

- 对于已激活分层存储的主题，Amazon MSK 不支持较小的日志段大小。如果要创建日志段，则最小段大小为 48MiB，或最短段滚动时间为 10 分钟。这些值映射到 segment.bytes 和 segment.ms 属性。
- local.retention.ms/bytes 的值不能等于或超过 retention.ms/bytes。这是分层存储保留设置。
- local.retention.ms/bytes 的默认值为 -2。这意味着 retention.ms 值用于 local.retention.ms/bytes。在这种情况下，数据将同时保留在本地存储和分层存储中（每个存储中各有一个副本），且会一起过期。对于此选项，本地数据的副本会保留到远程存储中。在这种情况下，从使用流量中读取的数据来自本地存储。
- retention.ms 的默认值为 7 天。retention.bytes 没有默认大小限制。
- retention.ms/bytes 的最小值为 -1。这意味着无限保留。
- local.retention.ms/bytes 的最小值为 -2。这意味着可以无限保留本地存储。它与 retention.ms/bytes 的设置 -1 相匹配。
- 对于已激活分层存储的主题，必须使用主题级别配置 retention.ms。最小 retention.ms 为 3 天。

Amazon MSK 配置操作

本主题说明如何创建自定义 MSK 配置以及如何对这些配置执行操作。有关如何使用 MSK 配置创建或更新集群的信息，请参阅 工作方式。

本主题包含下列部分：

- 创建 MSK 配置
- 更新 MSK 配置
- 删除 MSK 配置
- 描述 MSK 配置
- 描述 MSK 配置修订
- 列出您的账户中当前区域的所有 MSK 配置

创建 MSK 配置

1. 创建一个文件，可在其中指定要设置的配置属性以及要分配给这些属性的值。以下是示例配置文件的内容。

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

log.roll.ms = 604800000
```

2. 运行以下 AWS CLI 命令，并*config-file-path*替换为上一步中保存配置的文件的路径。

> **Note**
>
> 您为配置选择的名称必须符合以下正则表达式："^[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
```

以下是运行此命令后的成功响应示例。

```
{
```

```
    "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. 上一条命令会返回新配置的 Amazon 资源名称（ARN）。保存此 ARN，因为您需要使用它来在其他命令中引用此配置。如果您丢失了配置 ARN，则可列出账户中的所有配置来重新找到它。

更新 MSK 配置

1. 创建一个文件，可在其中指定要更新的配置属性以及要分配给这些属性的值。以下是示例配置文件的内容。

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

min.insync.replicas = 2
```

2. 运行以下 AWS CLI 命令，并*config-file-path*替换为上一步中保存配置的文件的路径。

   请将 *configuration-arn* 替换为您创建配置时获取的 ARN。如果您在创建配置时未保存 ARN，则可使用 `list-configurations` 命令列出账户中的所有配置。您想要在列表中显示的配置将显示在响应中。配置的 ARN 也将显示在该列表中。

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

3. 以下是运行此命令后的成功响应示例。

```
{
    "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
```

```
    }
}
```

删除 MSK 配置

以下程序展示如何删除未附加到集群的配置。您无法删除附加到集群的配置。

1. 要运行此示例，请将 *configuration-arn* 替换为您创建配置时获取的 ARN。如果您在创建配置时未保存 ARN，则可使用 `list-configurations` 命令列出账户中的所有配置。您想要在列表中显示的配置将显示在响应中。配置的 ARN 也将显示在该列表中。

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

2. 以下是运行此命令后的成功响应示例。

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

描述 MSK 配置

1. 以下命令会返回有关配置的元数据。要获取配置的详细说明，请运行 `describe-configuration-revision`。

   要运行此示例，请将 *configuration-arn* 替换为您创建配置时获取的 ARN。如果您在创建配置时未保存 ARN，则可使用 `list-configurations` 命令列出账户中的所有配置。您想要在列表中显示的配置将显示在响应中。配置的 ARN 也将显示在该列表中。

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

2. 以下是运行此命令后的成功响应示例。

```
{
    "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"
}
```

描述 MSK 配置修订

如果您使用 `describe-configuration` 命令描述 MSK 配置，您将看到配置的元数据。要获得配置的描述，请使用 `describe-configuration-revision` 命令。

- 运行以下命令，并将 *`configuration-arn`* 替换为您创建配置时获取的 ARN。如果您在创建配置时未保存 ARN，则可使用 `list-configurations` 命令列出账户中的所有配置。您想要在列表中显示的配置将显示在响应中。配置的 ARN 也将显示在该列表中。

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

  以下是运行此命令后的成功响应示例。

  ```
  {
      "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
  }
  ```

  `ServerProperties` 的值已使用 base64 进行编码。如果您使用 base64 解码器（例如 https://www.base64decode.org/）手动对其进行解码，则将获得用于创建自定义配置的原始配置文件的内容。在此情况下，您将获得以下内容：

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

```
log.roll.ms = 604800000
```

列出您的账户中当前区域的所有 MSK 配置

- 运行以下命令。

```
aws kafka list-configurations
```

以下是运行此命令后的成功响应示例。

```
{
    "Configurations": [
        {
            "Arn": "arn:aws:kafka:us-east-1:123456789012:configuration/SomeTest/
abcdabcd-abcd-1234-abcd-abcd123e8e8e-1",
            "CreationTime": "2019-05-21T00:54:23.591Z",
            "Description": "Example configuration description.",
            "KafkaVersions": [
                "1.1.1"
            ],
            "LatestRevision": {
                "CreationTime": "2019-05-21T00:54:23.591Z",
                "Description": "Example configuration description.",
                "Revision": 1
            },
            "Name": "SomeTest"
        },
        {
            "Arn": "arn:aws:kafka:us-east-1:123456789012:configuration/SomeTest/
abcdabcd-1234-abcd-1234-abcd123e8e8e-1",
            "CreationTime": "2019-05-03T23:08:29.446Z",
            "Description": "Example configuration description.",
            "KafkaVersions": [
                "1.1.1"
            ],
            "LatestRevision": {
                "CreationTime": "2019-05-03T23:08:29.446Z",
                "Description": "Example configuration description.",
                "Revision": 1
            },
```

```
            "Name": "ExampleConfigurationName"
        }
    ]
}
```

MSK Serverless

> **Note**
>
> MSK Serverless 在以下区域提供：美国东部（俄亥俄州）、美国东部（弗吉尼亚州北部）、美国西部（俄勒冈州）、加拿大（中部）、亚太地区（孟买）、亚太地区（新加坡）、亚太地区（悉尼）、亚太地区（东京）、亚太地区（首尔）、欧洲地区（法兰克福）、欧洲地区（斯德哥尔摩）、欧洲地区（爱尔兰）、欧洲地区（巴黎）和欧洲地区（伦敦）区域。

MSK Serverless 是 Amazon MSK 的一种集群类型，能让您无需管理和扩展集群容量即可运行 Apache Kafka。它可以在管理主题中的分区的同时自动配置和扩展容量，因此您可以流式传输数据，而无需考虑调整集群大小或扩展集群。MSK Serverless 提供基于吞吐量的定价模式，因此您只需为实际使用量付费。如果您的应用程序需要可自动向上和向下扩展的按需流式传输容量，请考虑使用无服务器集群。

MSK Serverless 与 Apache Kafka 完全兼容，因此您可以使用任何兼容的客户端应用程序来生成和使用数据。它还集成了以下服务：

- AWS PrivateLink，以提供私有连接
- AWS Identity and Access Management(IAM)，以使用 Java 和非 Java 语言进行身份验证和授权。有关为 IAM 配置客户端的说明，请参阅 配置客户端以进行 IAM 访问控制。
- AWS Glue 架构注册表，以进行架构管理
- 适用于 Apache Flink 的亚马逊托管服务，用于基于 Apache Flink 的流处理
- AWS Lambda，用于事件处理

> **Note**
>
> MSK Serverless 需要对所有集群进行 IAM 访问控制。不支持 Apache Kafka 访问控制列表（ACL）。有关更多信息，请参阅the section called “IAM 访问控制”。
> 有关适用于 MSK Serverless 的服务限额的信息，请参阅 the section called “无服务器集群的限额”。

为了帮助您开始使用无服务器集群，并详细了解无服务器集群的配置和监控选项，请参阅以下内容。

主题

开始使用 MSK Serverless 集群

本教程向您展示了一个示例，说明如何创建 MSK Serverless 集群，创建可以访问该集群的客户端，以及使用客户端在集群上创建主题并向这些主题写入数据。该练习并未提供您在创建无服务器集群时可以选择的所有选项。为了简单起见，我们在本练习的各个部分均选择默认选项。这并不意味着它们是可用于设置无服务器集群的唯一选项。您也可以使用 AWS CLI 或 Amazon MSK API。有关更多信息，请参阅 Amazon MSK API Reference 2.0。

主题

步骤 1：创建 MSK Serverless 集群

在此步骤中，您需执行两个任务。首先，使用默认设置创建一个 MSK Serverless 集群。然后，收集有关集群的信息。这是您在后续步骤中创建可向集群发送数据的客户端时所需的信息。

要创建无服务器集群

1. 登录 AWS Management Console 并打开 Amazon MSK 控制台，网址为 https://console.aws.amazon.com/msk/home。
2. 选择创建集群。
3. 对于创建方法，将快速创建选项保持为选中状态。快速创建选项允许您使用默认设置创建无服务器集群。
4. 对于集群名称，输入一个描述性名称，例如 **msk-serverless-tutorial-cluster**。
5. 对于常规集群属性，请选择无服务器作为集群类型。对于其余的常规集群属性，使用默认值。

6. 请注意所有集群设置下的表。此表列出了网络和可用性等重要设置的默认值，并指明了在创建集群后是否可以更改每项设置。要在创建集群之前更改设置，应在创建方法下选择自定义创建选项。

> **Note**
>
> 您最多可以将来自 5 个不同 VPC 的客户端与 MSK Serverless 集群连接。为了帮助客户端应用程序在发生中断时切换到另一个可用区，您必须在每个 VPC 中至少指定两个子网。

7. 选择创建集群。

要收集有关集群的信息

1. 在集群摘要部分，选择查看客户端信息。在 Amazon MSK 完成集群创建之前，此按钮将一直处于灰色状态。您可能需要等待几分钟直到按钮变为活动状态，然后才能使用。
2. 复制端点标签下的字符串。这是您的引导服务器字符串。
3. 选择 Properties（属性）选项卡。
4. 在网络设置部分下，复制子网和安全组的 ID 并保存，因为稍后需要这些信息来创建客户端计算机。
5. 选择任意子网。这将打开 Amazon VPC 控制台。查找与子网关联的 Amazon VPC 的 ID。保存此 Amazon VPC ID 以供将来使用。

下一步

步骤 2：创建 IAM 角色

步骤 2：创建 IAM 角色

在此步骤中，您需执行两个任务。第一个任务是创建 IAM policy，以授予在集群上创建主题以及向这些主题发送数据的访问权限。第二个任务是创建 IAM 角色并将此策略与其关联。在后面的步骤中，我们将创建代入此角色的客户端计算机，使用它在集群上创建主题并向该主题发送数据。

创建允许创建主题并写入主题的 IAM policy

1. 通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在导航窗格中，选择策略。
3. 请选择Create Policy（创建策略）。
4. 选择 JSON 选项卡，然后将编辑器窗口中的 JSON 替换为以下 JSON。

将 *region* 替换为您已创建集群的 AWS 区域 的代码。将 *Account-ID* 替换为您的账户 ID。将 *msk-serverless-tutorial-cluster* 替换为无服务器集群的名称。

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

```
}
```

有关如何写入安全策略的说明，请参阅 the section called “IAM 访问控制”。

5. 请选择下一步：标签。
6. 请选择下一步：审核。
7. 对于策略名称，输入一个描述性名称，例如 **msk-serverless-tutorial-policy**。
8. 选择 Create policy（创建策略）。

创建 IAM 角色并向其附加此策略

1. 在导航窗格中，选择角色。
2. 选择 Create role（创建角色）。
3. 在常见用例下，选择 EC2，然后选择下一步：权限。
4. 在搜索框中，输入您之前为本教程创建的策略的名称。然后，选中策略左侧的复选框。
5. 请选择下一步：标签。
6. 请选择下一步：审核。
7. 对于角色名称，输入一个描述性名称，例如 **msk-serverless-tutorial-role**。
8. 选择 Create role（创建角色）。

下一步

步骤 3：创建客户端计算机

步骤 3：创建客户端计算机

在此步骤中，您将执行两个任务。第一项任务是创建一个用作 Apache Kafka 客户端计算机的 Amazon EC2 实例。第二项任务是在计算机上安装 Java 和 Apache Kafka 工具。

创建客户端计算机

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 选择 Launch instance (启动实例)。
3. 为客户端计算机输入一个描述性名称，例如 **msk-serverless-tutorial-client**。
4. 对于亚马逊机器映像（AMI）类型，始终选中 Amazon Linux 2 AMI（HVM）– 内核 5.10，SSD 卷类型。

5. 保留 t2.micro 实例类型为选中状态。
6. 在密钥对（登录）下，选择创建新密钥对。对于密钥对名称，输入 **MSKServerlessKeyPair**。然后，选择 Download Key Pair (下载密钥对)。此外，您还可使用现有密钥对。
7. 对于网络设置，选择编辑。
8. 在 VPC 下，输入无服务器集群的虚拟私有云（VPC）的 ID。它是基于 Amazon VPC 服务的 VPC，您创建集群后保存了其 ID。
9. 对于子网，请选择您创建集群后保存了其 ID 的子网。
10. 在防火墙（安全组）中，选择与集群关联的安全组。如果该安全组有允许流量从安全组流向自身的入站规则，则此值有效。通过这样的规则，同一个安全组的成员可以相互通信。有关更多信息，请参阅《Amazon VPC 开发者指南》中的安全组规则。
11. 展开高级详细信息部分，然后选择您在 步骤 2：创建 IAM 角色 中创建的 IAM 角色。
12. 选择启动。
13. 在左侧导航窗格中，选择 Instances (实例)。然后选中代表您新创建的 Amazon EC2 实例的行中的复选框。从此时开始，我们称这个实例为客户端计算机。
14. 选择连接并按照说明连接到客户端计算机。

要在客户端计算机上设置 Apache Kafka 客户端工具

1. 要安装 Java，请在客户端计算机上运行以下命令：

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

2. 要获取创建主题和发送数据所需的 Apache Kafka 工具，请运行以下命令：

```
wget https://archive.apache.org/dist/kafka/2.8.1/kafka_2.12-2.8.1.tgz
```

```
tar -xzf kafka_2.12-2.8.1.tgz
```

3. 转到 `kafka_2.12-2.8.1/libs` 目录，然后运行以下命令以下载 Amazon MSK IAM JAR 文件。Amazon MSK IAM JAR 让客户端计算机可以访问集群。

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

4. 转到 `kafka_2.12-2.8.1/bin` 目录。复制以下属性设置并将其粘贴到新文件中。为文件 `client.properties` 命名并保存文件。

```
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
```

下一步

步骤 4：创建 Apache Kafka 主题

步骤 4：创建 Apache Kafka 主题

在此步骤中，您将使用先前创建的客户端计算机在无服务器集群上创建主题。

要创建主题并向其写入数据

1. 在以下 export 命令中，将 *my-endpoint* 替换为您在创建集群后保存的引导服务器字符串。然后，转到客户端计算机上的 kafka_2.12-2.8.1/bin 目录并运行 export 命令。

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

2. 运行以下命令以创建名为 msk-serverless-tutorial 的主题。

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

下一步

步骤 5：生成和使用数据

步骤 5：生成和使用数据

在此步骤中，您将使用在先前步骤中创建的主题生成和使用数据。

生成和使用消息

1. 运行以下命令以创建控制台生成器。

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

2. 输入所需的任何消息，然后按 Enter。重复执行此步骤两次或三次。每次输入一行并按 Enter 时，该行会作为单独的消息发送到集群。
3. 将与客户端计算机的连接保持打开状态，然后在新窗口中打开与该计算机的第二个单独连接。
4. 使用客户端计算机的第二个连接，通过以下命令创建控制台使用器。将 *my-endpoint* 替换为您在创建集群后保存的引导服务器字符串。

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

   您开始看到之前使用控制台生成器命令时输入的消息。
5. 在生成器窗口中输入更多消息，并观察消息显示在使用器窗口中。

下一步

步骤 6：删除资源

步骤 6：删除资源

在此步骤中，您将删除在本教程中创建的资源。

要删除集群

1. 通过以下网址打开 Amazon MSK 控制台：https://console.aws.amazon.com/msk/home。
2. 在集群列表中，选择为此教程创建的集群。
3. 对于操作，选择删除集群。
4. 在字段中输入 `delete`，然后选择删除。

要停止客户端计算机

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在 Amazon EC2 实例列表中，选择您为本教程创建的客户端计算机。
3. 选择实例状态，然后选择终止实例。

4. 选择 Terminate (终止)。

删除 IAM policy 和角色

1. 通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在导航窗格中，选择角色。
3. 在搜索框中，输入您为本教程创建的 IAM 角色的名称。
4. 选择角色。然后选择删除角色并确认删除。
5. 在导航窗格中，选择策略。
6. 在搜索框中，输入您为本教程创建的策略的名称。
7. 选择策略，打开其摘要页面。在策略的摘要页面上，选择删除策略。
8. 选择 Delete。

无服务器集群的配置

Amazon MSK 会为无服务器集群设置代理配置属性。您无法更改这些代理配置属性设置。不过，您可以设置以下主题配置属性。

配置属性	默认值	可编辑	允许的最大值
cleanup.policy	Delete	是，但仅限于主题创建时	
compression.type	Producer	是	
max.message.bytes	1048588	是	8MiB
message.timestamp.difference.max.ms	long.max	是	
message.timestamp.type	CreateTime	是	
retention.bytes	250GiB	是	250GiB
retention.ms	7 days	是	无限

您也可以使用 Apache Kafka 命令为新主题或现有主题设置或修改主题级别的配置属性。有关主题级别的配置属性以及如何设置这些属性的示例的更多信息，请参阅 Apache Kafka 文档中的 Topic-Level Configs。

监控无服务器集群

Amazon MSK 与 Amazon CloudWatch 集成，因此您可以收集、查看和分析 MSK Serverless 集群的指标。下表所示为适用于所有无服务器集群的指标。由于这些指标是作为主题中每个分区的单独数据点发布的，因此我们建议将它们作为“SUM”统计数据进行查看，以获得主题级别的视图。

Amazon MSK 以每分钟一次的频率向 CloudWatch 发布 PerSec 指标。这意味着，一分钟的“SUM”统计数据可以准确地表示 PerSec 指标的每秒数据。要收集超过一分钟的时间段的每秒数据，请使用以下 CloudWatch 数学表达式：m1 * 60/PERIOD(m1)。

默认监控级别可用的指标

名称	可见时间	Dimensions	描述
BytesInPerSec	在生成器写入主题之后	集群名称、主题	每秒从客户端接收的字节数。此指标对每个主题都可用。
BytesOutPerSec	在使用器组使用某个主题之后。	集群名称、主题	每秒发送到客户端的字节数。此指标对每个主题都可用。
FetchMessageConversionsPerSec	在使用器组使用某个主题之后。	集群名称、主题	主题每秒提取消息转换的次数。
EstimatedMaxTimeLag	在使用器组使用某个主题之后。	集群名称、使用器组、主题	MaxOffsetLag 指标的时间估计值。
MaxOffsetLag	在使用器组使用某个主题之后。	集群名称、使用器组、主题	主题中所有分区之间的最大偏移延迟。
MessagesInPerSec	在生成器写入主题之后	集群名称、主题	主题每秒传入消息数。
ProduceMessageConversionsPerSec	在生成器写入主题之后	集群名称、主题	主题每秒生成的消息转换数。

名称	可见时间	Dimensions	描述
SumOffsetLag	在使用器组使用某个主题之后。	集群名称、使用器组、主题	主题中所有分区的聚合偏移延迟。

要查看 MSK Serverless 指标

1. 登录AWS Management Console并打开 CloudWatch 控制台，网址为 https://console.aws.amazon.com/cloudwatch/。
2. 在导航窗格中，依次选择指标、所有指标。
3. 在指标中搜索 **kafka** 一词。
4. 选择 AWS/Kafka/集群名称、主题或 AWS/Kafka/集群名称、使用器组、主题以查看不同的指标。

MSK Connect

什么是 MSK Connect？

MSK Connect 是 Amazon MSK 的一项功能，它让开发人员可以轻松地将数据流入和流出其 Apache Kafka 集群。MSK Connect 使用 Kafka Connect 2.7.1，这是一个开源框架，用于将 Apache Kafka 集群与数据库、搜索索引和文件系统等外部系统连接起来。借助 MSK Connect，您可以部署专为 Kafka Connect 构建的完全托管的连接器，用于将数据移入亚马逊 S3 和亚马逊服务等热门数据存储或从中提取数据。 OpenSearch 您可以部署由 Debezium 等第三方开发的连接器，用于将变更日志从数据库流式传输到 Apache Kafka 集群，或者无需更改代码即可部署现有连接器。连接器会自动扩缩以适应负载变化，您只需为使用的资源付费。

使用源连接器将数据从外部系统导入到您的主题中。您可以使用接收器连接器，将主题中的数据导出到外部系统。

MSK Connect 支持任何连接到 Amazon VPC 的 Apache Kafka 集群的连接器，无论是 MSK 集群还是独立托管的 Apache Kafka 集群。

MSK Connect 持续监控连接器的运行状况和交付状态、修补和管理底层硬件，并自动扩缩连接器以适应吞吐量的变化。

要开始使用 MSK Connect，请参阅 the section called “开始使用”。

要了解您可以使用 MSK Connect 创建的 AWS 资源，请参阅 the section called “连接器”、the section called “插件” 和 the section called “工作线程”。

有关 MSK Connect API 的信息，请参阅 Amazon MSK Connect API Reference。

开始使用 MSK Connect

这是一个分步教程，使用 AWS Management Console 创建 MSK 集群和将数据从集群发送到 S3 存储桶的接收器连接器。

主题

- 步骤 1：设置所需资源
- 步骤 2：创建自定义插件

步骤 1：设置所需资源

在此步骤中，您需创建此入门场景所需的以下资源：

- 一个 S3 存储桶，用作从连接器接收数据的目的地。
- 一个 MSK 集群，您将向其发送数据。然后，连接器将从此集群读取数据并将其发送到目标 S3 存储桶。
- 一个 IAM 角色，允许连接器写入目标 S3 存储桶。
- 一个 Amazon VPC 端点，可以将数据从具有集群和连接器的 Amazon VPC 发送到 Amazon S3。

创建 S3 存储桶

1. 登录到 AWS Management Console，然后通过以下网址打开 Amazon S3 控制台：https://console.aws.amazon.com/s3/。
2. 请选择 Create bucket（创建桶）。
3. 对于存储桶名称，输入一个描述性名称，例如 `mkc-tutorial-destination-bucket`。
4. 向下滚动并选择创建存储桶。
5. 在存储桶列表中，选择您新创建的存储桶。
6. 请选择 Create folder（创建文件夹）。
7. 输入 `tutorial` 作为文件夹的名称，然后向下滚动并选择创建文件夹。

创建集群

1. 通过以下网址打开 Amazon MSK 控制台：https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。
2. 在左侧窗格的 MSK 集群下，选择集群。
3. 选择创建集群。
4. 选择自定义创建。
5. 对于集群名称，请输入 `mkc-tutorial-cluster`。

6. 在“常规集群属性”下，为集群类型选择已预置。
7. 在网络下，选择“Amazon VPC”。然后选择想要使用的可用区和子网。记住您选择的 Amazon VPC 和子网的 ID，因为您将在本教程的后面部分需要它们。
8. 在访问控制方法下，确保仅选择未经身份验证的访问。
9. 在加密下，确保仅选择明文。
10. 继续执行向导，然后选择创建集群。这会将您引导至该集群的“详细信息”页面。在该页面的已应用的安全组下，找到安全组 ID。记住该 ID，因为您将在本教程的后面部分需要它。

创建可以写入目标存储桶的 IAM 角色

1. 通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在左侧窗格的访问管理下，选择角色。
3. 选择 Create role（创建角色）。
4. 在或选择一个服务以查看其用例中，选择 S3。
5. 向下滚动，在选择您的用例下，再次选择 S3。
6. 请选择下一步: 权限。
7. 选择 Create policy（创建策略）。这将在您的浏览器中打开一个新选项卡，您需在其中创建策略。保持原始角色创建选项卡处于打开状态，因为我们稍后将返回到该选项卡。
8. 选择 JSON 选项卡，然后将窗口中的文本替换为以下策略。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets"
      ],
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation",
        "s3:DeleteObject"
```

```
      ],
      "Resource": "arn:aws:s3:::<my-tutorial-destination-bucket>"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:AbortMultipartUpload",
        "s3:ListMultipartUploadParts",
        "s3:ListBucketMultipartUploads"
      ],
      "Resource": "*"
    }
  ]
}
```

9. 请选择下一步：标签。
10. 请选择下一步：审核。
11. 输入 `mkc-tutorial-policy` 作为策略名称，然后向下滚动并选择创建策略。
12. 返回创建角色的浏览器选项卡，选择刷新按钮。
13. 找到 `mkc-tutorial-policy` 并通过选择其左侧的按钮将其选中。
14. 请选择下一步：标签。
15. 请选择下一步：审核。
16. 输入 `mkc-tutorial-role` 作为角色名称，然后删除描述框中的文本。
17. 选择 Create role（创建角色）。

允许 MSK Connect 代入该角色

1. 在 IAM 控制台的左侧窗格中，在访问管理下，选择角色。
2. 找到 `mkc-tutorial-role` 并将其选中。
3. 在角色的摘要下，选择信任关系选项卡。
4. 选择 Edit trust relationship (编辑信任关系)。
5. 将现有信任策略替换为以下 JSON。

```
{
```

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

6. 选择 Update Trust Policy（更新信任策略）。

创建从集群的 VPC 到 Amazon S3 的 Amazon VPC 端点

1. 通过以下网址打开 Amazon VPC 控制台：https://console.aws.amazon.com/vpc/。
2. 在左侧窗格中，选择端点。
3. 选择 Create endpoint（创建端点）。
4. 在服务名称下，选择 com.amazonaws.us-east-1.s3 服务和网关类型。
5. 选择集群的 VPC，然后选中与集群子网关联的路由表左侧的复选框。
6. 选择 Create endpoint（创建端点）。

下一步

步骤 2：创建自定义插件

步骤 2：创建自定义插件

插件包含定义连接器逻辑的代码。在此步骤中，您需创建一个包含 Lenses Amazon S3 接收器连接器代码的自定义插件。在后面的步骤中，当您创建 MSK 连接器时，您可以指定其代码位于此自定义插件中。您可以使用同一插件来创建多个具有不同配置的 MSK 连接器。

创建自定义插件

1. 下载 S3 连接器。
2. 将 ZIP 文件上传到您有权访问的 S3 存储桶。有关如何将文件上传到 Amazon S3 的信息，请参阅《Amazon S3 用户指南》中的上传对象。

3. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
4. 在左侧窗格中展开 MSK Connect，然后选择自定义插件。
5. 选择创建自定义插件。
6. 选择 Browse S3（浏览 S3）。
7. 在存储桶列表中，找到您上传 ZIP 文件的存储桶，然后选择该存储桶。
8. 在存储桶的对象列表中，选择 ZIP 文件左侧的单选按钮，然后选择标有选择的按钮。
9. 输入 `mkc-tutorial-plugin` 作为自定义插件名称，然后选择创建自定义插件。

AWS 可能需要几分钟才能完成自定义插件的创建。创建过程完成后，您会在浏览器窗口顶部的横幅中看到以下消息。

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

下一步

步骤 3：创建客户端计算机和 Apache Kafka 主题

步骤 3：创建客户端计算机和 Apache Kafka 主题

在此步骤中，您需创建 Amazon EC2 实例以用作 Apache Kafka 客户端实例。然后，您可以使用此实例在集群上创建主题。

创建客户端计算机

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 选择 Launch instances。
3. 输入客户端计算机的名称，例如 **`mkc-tutorial-client`**。
4. 对于亚马逊机器映像（AMI）类型，始终选中 Amazon Linux 2 AMI（HVM）– 内核 5.10，SSD 卷类型。
5. 选择 t2.xlarge 实例类型。
6. 在密钥对（登录）下，选择创建新密钥对。为密钥对名称输入 **`mkc-tutorial-key-pair`**，然后选择下载密钥对。此外，您还可使用现有密钥对。
7. 选择 Launch instance (启动实例)。

8. 选择查看实例。然后，在安全组列中，选择与新的实例关联的安全组。复制并保存安全组的 ID，以供稍后使用。

允许新创建的客户端向集群发送数据

1. 通过以下网址打开 Amazon VPC 控制台：https://console.aws.amazon.com/vpc/。
2. 在左侧窗格的安全性下，选择安全组。在安全组 ID 列中，找到集群的安全组。您在 the section called “步骤 1：设置所需资源” 中创建集群时保存了该安全组的 ID。通过选中该安全组行左侧的复选框来选择该安全组。确保没有同时选择其他安全组。
3. 在屏幕的下半部分，选择入站规则选项卡。
4. 选择 Edit inbound rules（编辑入站规则）。
5. 在屏幕的左下角，选择添加规则。
6. 在新规则中，选择类型列中的所有流量。在源列右侧的字段中，输入客户端计算机的安全组 ID。这是您在创建客户端计算机后保存的安全组 ID。
7. 选择 Save rules（保存规则）。您的 MSK 集群现在将接受来自您在上一程序中创建的客户端的所有流量。

要创建主题，请执行以下操作

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在实例表中选择 `mkc-tutorial-client`。
3. 在屏幕顶部附近，选择连接，然后按照说明连接到实例。
4. 通过运行以下命令在客户端实例上安装 Java：

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

5. 运行以下命令以下载 Apache Kafka。

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

> **Note**
>
> 如果您希望使用此命令中使用的镜像站点之外的镜像站点，则可在 Apache 网站上选择其他镜像站点。

6. 在上一步中将 TAR 文件下载到的目录中运行以下命令。

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

7. 转到 kafka_2.12-2.2.1 目录。
8. 通过以下网址打开 Amazon MSK 控制台：https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。
9. 在左侧窗格中，选择集群，然后选择名称 mkc-tutorial-cluster。
10. 选择查看客户端信息。
11. 复制明文连接字符串。
12. 选择完成。
13. 在客户端实例 (mkc-tutorial-client) 上运行以下命令，并将 *bootstrapServerString* 替换为您在查看集群客户端信息时保存的值。

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

如果此命令成功，您将看到以下消息：Created topic mkc-tutorial-topic.

下一步

步骤 4：创建连接器

步骤 4：创建连接器

创建连接器

1. 登录 AWS Management Console 并通过以下网址打开 Amazon MSK 控制台：https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。
2. 在左侧窗格中，展开 MSK Connect，然后选择连接器。
3. 选择 Create connector (创建连接器)。
4. 在插件列表中，选择 mkc-tutorial-plugin，然后选择下一步。
5. 对于连接器名称，请输入 mkc-tutorial-connector。
6. 在集群列表中，选择 mkc-tutorial-cluster。
7. 复制以下配置，并将其粘贴到连接器配置字段中。

```
connector.class=io.confluent.connect.s3.S3SinkConnector
s3.region=us-east-1
format.class=io.confluent.connect.s3.format.json.JsonFormat
flush.size=1
schema.compatibility=NONE
tasks.max=2
topics=mkc-tutorial-topic
partitioner.class=io.confluent.connect.storage.partitioner.DefaultPartitioner
storage.class=io.confluent.connect.s3.storage.S3Storage
s3.bucket.name=<my-tutorial-destination-bucket>
topics.dir=tutorial
```

8. 在访问权限下，选择 mkc-tutorial-role。
9. 选择 Next（下一步）。在安全性页面上，再次选择下一步。
10. 在日志页面上，选择下一步。
11. 在查看并创建下，选择创建连接器。

下一步

步骤 5：发送数据

步骤 5：发送数据

在此步骤中，您将数据发送到之前创建的 Apache Kafka 主题，然后在目标 S3 存储桶中查找相同的数据。

向 MSK 集群发送数据

1. 在客户端实例上的 Apache Kafka 安装 bin 文件夹中，创建一个名为 client.properties 的文本文件，该文件包含以下内容。

```
security.protocol=PLAINTEXT
```

2. 运行以下命令以创建控制台生成器。将 *BootstrapBrokerString* 替换为您运行上一条命令时获得的值。

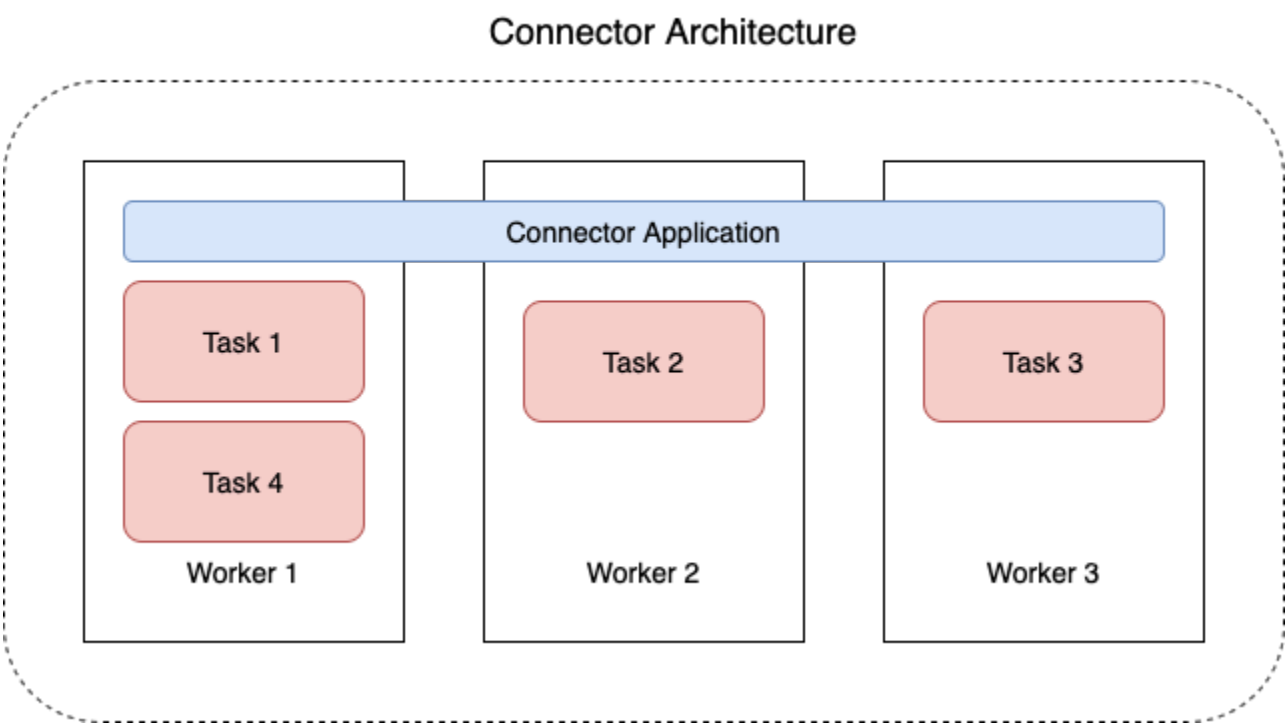
```
<path-to-your-kafka-installation>/bin/kafka-console-producer.sh --broker-
list BootstrapBrokerString --producer.config client.properties --topic mkc-
tutorial-topic
```

3. 输入所需的任何消息，然后按 Enter。重复执行此步骤两次或三次。每次输入一行并按 Enter 时，该行会作为单独的消息发送到您的 Apache Kafka 集群。
4. 查看目标 Amazon S3 存储桶，查找您在上一步中发送的消息。

连接器

连接器会持续将数据来源中的流数据复制到您的 Apache Kafka 集群，或者持续将数据从集群复制到数据接收器中，从而将外部系统和 Amazon 服务与 Apache Kafka 集群相集成。连接器还可以执行轻量级逻辑，例如在将数据传送到目标之前进行转换、格式转换或数据筛选。源连接器从数据来源提取数据，并将这些数据推送到集群中，而接收器连接器则从集群中提取数据，并将这些数据推送到数据接收器中。

下图显示了连接器的架构。工作程序是运行连接器逻辑的 Java 虚拟机（JVM）进程。每个工作程序都会创建一组任务，这些任务在并行线程中运行并执行复制数据的工作。任务不存储状态，因此可以随时启动、停止或重新启动，以提供弹性且可扩展的数据管道。



连接器容量

连接器的总容量取决于该连接器拥有的工作程序数量，以及每个工作程序的 MSK Connect 单位（MCU）数量。每个 MCU 代表 1 个 vCPU 的计算能力和 4GiB 的内存。MCU 内存与工作程序实例的总内存有关，而不是正在使用的堆内存。

MSK Connect 工作人员使用客户提供的子网中的 IP 地址。每个工作人员使用客户提供的子网中的一个 IP 地址。您应确保在提供给 CreateConnector 请求的子网中有足够的可用的 IP 地址来考虑其指定容量，尤其是在自动缩放连接器时，工作人员数量可能会波动。

要创建连接器，必须选择以下两种容量模式之一。

- 已预置 – 如果您知道连接器的容量要求，请选择此模式。指定两个值：
  - 工作程序数量。
  - 每个工作程序的 MCU 数量。
- 自动扩缩 – 如果连接器的容量要求各不相同，或者您事先不知道连接器的容量要求，请选择此模式。当您使用自动扩缩模式时，Amazon MSK Connect 会覆盖连接器的 `tasks.max` 属性，其值与连接器中运行的工作程序数量和每个工作程序的 MCU 数量成正比。

  指定三组值：
  - 最小和最大工作程序数量。
  - CPU 利用率的横向缩减百分比和横向扩展百分比，该百分比由 `CpuUtilization` 指标确定。当连接器的 `CpuUtilization` 指标超过横向扩展百分比时，MSK Connect 会增加连接器中运行的工作程序数量。当 `CpuUtilization` 指标低于横向缩减百分比时，MSK Connect 会减少工作程序数量。工作程序的数量将始终保持在创建连接器时指定的最小和最大数量之间。
  - 每个工作程序的 MCU 数量。

有关工作程序的更多信息，请参阅 the section called “工作线程”。要了解有关 MSK Connect 指标的信息，请参阅 the section called “监控”。

创建连接器

使用创建连接器 AWS Management Console

1. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
2. 在左侧窗格的 MSK Connect 下，选择连接器。
3. 选择 Create connector (创建连接器)。
4. 您可以选择使用现有的自定义插件来创建连接器，也可以先创建新的自定义插件。有关自定义插件以及如何创建这些插件的信息，请参阅 the section called “插件”。在此过程中，假设您有一个要使用的自定义插件。在自定义插件列表中，找到要使用的插件，选中其左侧的复选框，然后选择下一步。
5. 输入名称和描述（可选）。

6. 选择您想要连接到的集群。
7. 指定连接器配置。您需要指定的配置参数取决于要创建的连接器类型。但是，部分参数是所有连接器通用的参数，例如 `connector.class` 和 `tasks.max` 参数。以下是 Confluent Amazon S3 Sink Connector 的配置示例。

```
connector.class=io.confluent.connect.s3.S3SinkConnector
tasks.max=2
topics=my-example-topic
s3.region=us-east-1
s3.bucket.name=my-destination-bucket
flush.size=1
storage.class=io.confluent.connect.s3.storage.S3Storage
format.class=io.confluent.connect.s3.format.json.JsonFormat
partitioner.class=io.confluent.connect.storage.partitioner.DefaultPartitioner
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.kafka.connect.storage.StringConverter
schema.compatibility=NONE
```

8. 接下来，配置您的连接器容量。您可以在两种容量模式之间选择：已预置和自动扩缩。有关这两个选项的信息，请参阅the section called “容量”。
9. 选择默认工作程序配置或自定义工作程序配置。有关创建自定义工作程序配置的信息，请参阅 the section called “工作线程”。
10. 接下来，指定服务执行角色。这必须是 MSK Connect 可以担任的 IAM 角色，该角色向连接器授予访问必要 AWS 资源所需的所有权限。这些权限取决于连接器的逻辑。有关如何创建此角色的信息，请参阅 the section called “服务执行角色”。
11. 选择下一步，查看安全信息，然后再次选择下一步。
12. 指定所需的日志记录选项，然后选择下一步。有关日志记录的信息，请参阅the section called “日志记录”。
13. 选择 Create connector (创建连接器)。

要使用 MSK Connect API 创建连接器，请参阅CreateConnector。

插件

插件是一种 AWS 资源，其中包含的代码可定义您的连接器逻辑。您可以将 JAR 文件（或包含一个或多个 JAR 文件的 ZIP 文件）上传到 S3 存储桶，并在创建插件时指定存储桶的位置。创建连接器时，

需要指定您想要 MSK Connect 用于该连接器的插件。插件与连接器的关系是一对多的：您可以从同一个插件创建一个或多个连接器。

有关如何开发连接器代码的信息，请参阅 Apache Kafka 文档中的连接器开发指南。

使用 AWS Management Console 创建自定义插件

1. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
2. 在左侧窗格的 MSK Connect 下，选择自定义插件。
3. 选择创建自定义插件。
4. 选择 Browse S3（浏览 S3）。
5. 在 S3 存储桶列表中，选择包含插件的 JAR 或 ZIP 文件的存储桶。
6. 在对象列表中，选中插件的 JAR 或 ZIP 文件左侧的复选框，然后选择选择。
7. 选择创建自定义插件。

要使用 MSK Connect API 创建自定义插件，请参阅 CreateCustomPlugin。

工作线程

工作程序是运行连接器逻辑的 Java 虚拟机（JVM）进程。每个工作程序都会创建一组任务，这些任务在并行线程中运行并执行复制数据的工作。任务不存储状态，因此可以随时启动、停止或重新启动，以提供弹性且可扩展的数据管道。剩余工作程序会自动检测到工作程序数量的变化，无论是由于扩展事件还是意外故障所致。它们会进行协调，以重新平衡剩余工作程序集合的任务。Connect 工作程序使用 Apache Kafka 的使用器组来协调和重新平衡。

如果您的连接器容量要求变化不定或难以估计，则可以让 MSK Connect 根据需要在您指定的下限和上限之间扩展工作程序数量。或者，您可以指定要运行连接器逻辑的确切工作程序数量。有关更多信息，请参阅the section called “容量”。

MSK Connect 工作人员消耗 IP 地址

MSK Connect 工作人员使用客户提供的子网中的 IP 地址。每个工作人员使用客户提供的子网中的一个 IP 地址。您应确保在提供给 CreateConnector 请求的子网中有足够的可用的 IP 地址来考虑其指定容量，尤其是在自动缩放连接器时，工作人员数量可能会波动。

主题

- 默认工作程序配置

- 支持的工作程序配置属性
- 创建自定义工作程序配置
- 使用 offset.storage.topic 管理源连接器偏移

默认工作程序配置

MSK Connect 提供以下默认工作程序配置：

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

支持的工作程序配置属性

MSK Connect 提供默认的工作程序配置。您还可以选择创建用于连接器的自定义工作程序配置。以下列表包含有关 Amazon MSK Connect 支持或不支持的工作程序配置属性的信息。

- 只有 `key.converter` 和 `value.converter` 属性为必需。
- MSK Connect 支持以下 `producer.` 配置属性。

```
producer.acks
producer.batch.size
producer.buffer.memory
producer.compression.type
producer.enable.idempotence
producer.key.serializer
producer.max.request.size
producer.metadata.max.age.ms
producer.metadata.max.idle.ms
producer.partitioner.class
producer.reconnect.backoff.max.ms
producer.reconnect.backoff.ms
producer.request.timeout.ms
producer.retry.backoff.ms
producer.value.serializer
```

- MSK Connect 支持以下 `consumer.` 配置属性。

```
consumer.allow.auto.create.topics
consumer.auto.offset.reset
consumer.check.crcs
```

```
consumer.fetch.max.bytes
consumer.fetch.max.wait.ms
consumer.fetch.min.bytes
consumer.heartbeat.interval.ms
consumer.key.deserializer
consumer.max.partition.fetch.bytes
consumer.max.poll.records
consumer.metadata.max.age.ms
consumer.partition.assignment.strategy
consumer.reconnect.backoff.max.ms
consumer.reconnect.backoff.ms
consumer.request.timeout.ms
consumer.retry.backoff.ms
consumer.session.timeout.ms
consumer.value.deserializer
```

- 支持所有其他不以 `producer.` 或 `consumer.` 前缀开头的配置属性，但以下属性除外。

```
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
```

有关工作程序配置属性及其表示的更多信息，请参阅 Apache Kafka 文档中的 Kafka Connect Configs。

创建自定义工作程序配置

使用创建自定义工作器配置 AWS Management Console

1. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
2. 在左侧窗格的 MSK Connect 下，选择工作程序配置。
3. 选择创建工作程序配置。
4. 输入名称和可选描述，然后添加属性和要将属性设置为的值。
5. 选择创建工作程序配置。

要使用 MSK Connect API 创建工作器配置，请参阅CreateWorkerConfiguration。

使用 `offset.storage.topic` 管理源连接器偏移

本节提供的信息可帮助您使用偏移存储主题管理源连接器偏移。偏移存储主题是 Kafka Connect 用来存储连接器和任务配置偏移的内部主题。

使用默认偏移存储主题

默认情况下，Amazon MSK Connect 会在 Kafka 集群上为您创建的每个连接器生成一个新的偏移存储主题。MSK 使用部分连接器 ARN 构造默认主题名称。例如，`__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-4abc-8be8-c657f7e4ff32-2`。

指定您自己的偏移存储主题

要在源连接器之间提供偏移连续性，您可以使用自己选择的偏移存储主题来代替默认主题。指定偏移存储主题可以帮助您完成创建源连接器之类的任务，该连接器可从上一个连接器的最后一个偏移恢复读取。

要指定偏移存储主题，请在创建连接器之前在工作程序配置中为 `offset.storage.topic` 属性提供一个值。如果要重复使用偏移存储主题来消耗先前创建的连接器的偏移，则必须为新连接器指定与旧连接器相同的名称。如果您创建自定义偏移存储主题，则必须在主题配置中将 `cleanup.policy` 设置为 `compact`。

> **Note**
>
> 如果您在创建接收器连接器时指定了偏移存储主题，若该主题尚不存在，则 MSK Connect 会创建该主题。但是，该主题不会用于存储连接器偏移，
> 而是使用 Kafka 使用器组协议来管理接收器连接器偏移。每个接收器连接器都会创建一个名为 `connect-{CONNECTOR_NAME}` 的组。只要使用器组存在，您创建的任何具有相同 `CONNECTOR_NAME` 值的连续接收器连接器都将从上次提交的偏移继续。

Example ：指定偏移存储主题以使用更新后的配置重新创建源连接器

假设您有一个更改数据捕获（CDC）连接器，并且您想在不丢失 CDC 流中的位置的情况下修改连接器配置。您无法更新现有的连接器配置，但可以删除连接器并使用相同的名称创建新连接器。要告诉新连接器在 CDC 流中从何处开始读取，您可以在工作程序配置中指定旧连接器的偏移存储主题。以下步骤演示如何完成此任务。

1. 在您的客户端计算机上，运行以下命令以查找连接器偏移存储主题的名称。将 *<bootstrapBrokerString>* 替换为集群的引导代理字符串。有关获取引导代理字符串的说明，请参阅 获取 Amazon MSK 集群的引导代理。

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

   以下输出显示了所有集群主题的列表，包括所有默认的内部连接器主题。在此示例中，现有 CDC 连接器使用由 MSK Connect 创建的默认偏移存储主题。这就是偏移存储主题名为 `__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-4abc-8be8-c657f7e4ff32-2` 的原因。

   ```
   __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. 通过以下网址打开 Amazon MSK 控制台：https://console.aws.amazon.com/msk/。
3. 从连接器列表中选择您的连接器。复制并保存连接器配置字段的内容，以便您可以对其进行修改并使用它来创建新连接器。
4. 要删除连接器，请选择删除。然后在文本输入字段中输入连接器名称，以确认删除。
5. 使用适合您场景的值创建自定义工作程序配置。有关说明，请参阅创建自定义工作程序配置。

   在工作程序配置中，必须将之前检索到的偏移存储主题的名称指定为类似于以下配置中 `offset.storage.topic` 的值。

```
config.providers.secretManager.param.aws.region=us-east-1
key.converter=<org.apache.kafka.connect.storage.StringConverter>
value.converter=<org.apache.kafka.connect.storage.StringConverter>
config.providers.secretManager.class=com.github.jcustenborder.kafka.config.aws.SecretsManag
config.providers=secretManager
offset.storage.topic=__amazon_msk_connect_offsets_my-mskc-
connector_12345678-09e7-4abc-8be8-c657f7e4ff32-2
```

6. **Important**

   必须为新连接器指定与旧连接器相同的名称。

   使用在上一步中设置的工作程序配置创建新连接器。有关说明，请参阅创建连接器。

注意事项

在管理源连接器偏移时，请考虑以下几点。

- 要指定偏移存储主题，请提供将连接器偏移作为工作程序配置中 `offset.storage.topic` 的值进行存储的 Kafka 主题名称。
- 更改连接器配置时要谨慎行事。如果源连接器将配置中的值用于键偏移记录，则更改配置值可能会导致连接器出现意想不到的行为。我们建议您参考插件的文档以获取指导。
- 自定义默认分区数 – 除了通过添加 `offset.storage.topic` 来自定义工作程序配置外，您还可以为偏移和状态存储主题自定义分区数量。内部主题的默认分区如下。

  - `config.storage.topic`：1，不可配置，必须是单分区主题
  - `offset.storage.topic`：25，可通过提供 `offset.storage.partitions` 进行配置
  - `status.storage.topic`：5，可通过提供 `status.storage.partitions` 进行配置
- 手动删除主题 – Amazon MSK Connect 在每次部署连接器时都会创建新的 Kafka 连接内部主题（主题名称以 `__amazon_msk_connect` 开头）。附加到已删除连接器的旧主题不会自动删除，因为内部主题（例如 `offset.storage.topic`）可以在连接器之间重复使用。但是，您可以手动删除 MSK Connect 创建的未使用的内部主题。内部主题按照 `__amazon_msk_connect_<offsets|status|configs>_`*`connector_name_connector_id`* 格式命名。

  正则表达式 `__amazon_msk_connect_<offsets|status|configs>_`*`connector_name_connector_id`* 可用于删除内部主题。您不应删除正在运行的连接器当前正在使用的内部主题。
- 对 MSK Connect 创建的内部主题使用相同名称 – 如果要重复使用偏移存储主题来消耗先前创建的连接器的偏移，则必须为新连接器指定与旧连接器相同的名称。可以使用工作程序配置来设置 `offset.storage.topic` 属性，以便将相同的名称分配到 `offset.storage.topic`，并在不同的连接器之间重复使用。管理连接器偏移中描述了此配置。MSK Connect 不允许不同的连接器共享 `config.storage.topic` 和 `status.storage.topic`。每次在 MSKC 中创建新连接器时都会创建这些主题。它们会按照 `__amazon_msk_connect_<status|configs>_`*`connector_name_connector_id`* 格式自动命名，因此在您创建的不同连接器中会有所不同。

使用配置提供程序将敏感信息外部化

此示例演示了如何使用开源配置提供程序将 Amazon MSK Connect 的敏感信息外部化。配置提供程序允许您在连接器或工作程序配置中指定变量而不是明文，在连接器中运行的工作程序会在运行时系统解析这些变量。这样可以防止凭证和其他密钥以明文形式存储。示例中的配置提供程序支持从 AWS Secrets Manager、Amazon S3 和 Systems Manager（SSM）检索配置参数。在步骤 2 中，您可以看到如何为要配置的服务设置敏感信息的存储和检索。

主题

步骤 1：创建自定义插件并上传到 S3

要创建自定义插件，可通过在本地计算机上运行以下命令来创建包含连接器和 msk-config-provider 的 zip 文件。

使用终端窗口和 Debezium 作为连接器创建自定义插件

使用 AWS CLI 以拥有允许您访问 AWS S3 存储桶的凭证的超级用户身份运行命令。有关安装和设置 AWS CLI 的信息，请参阅《AWS Command Line Interface 用户指南》中的开始使用 AWS CLI。有关将 AWS CLI 与 Amazon S3 结合使用的信息，请参阅《AWS Command Line Interface 用户指南》中的将 Amazon S3 与 AWS CLI 结合使用。

1. 在终端窗口中，使用以下命令在工作区中创建一个名为 `custom-plugin` 的文件夹。

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

2. 使用以下命令从 Debezium 网站下载最新稳定版本的 MySQL Connector 插件。

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

   使用以下命令将下载的 gzip 文件提取到 `custom-plugin` 文件夹中。

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

3. 使用以下命令下载 MSK 配置提供程序 zip 文件。

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

   使用以下命令将下载的 zip 文件提取到 `custom-plugin` 文件夹中。

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

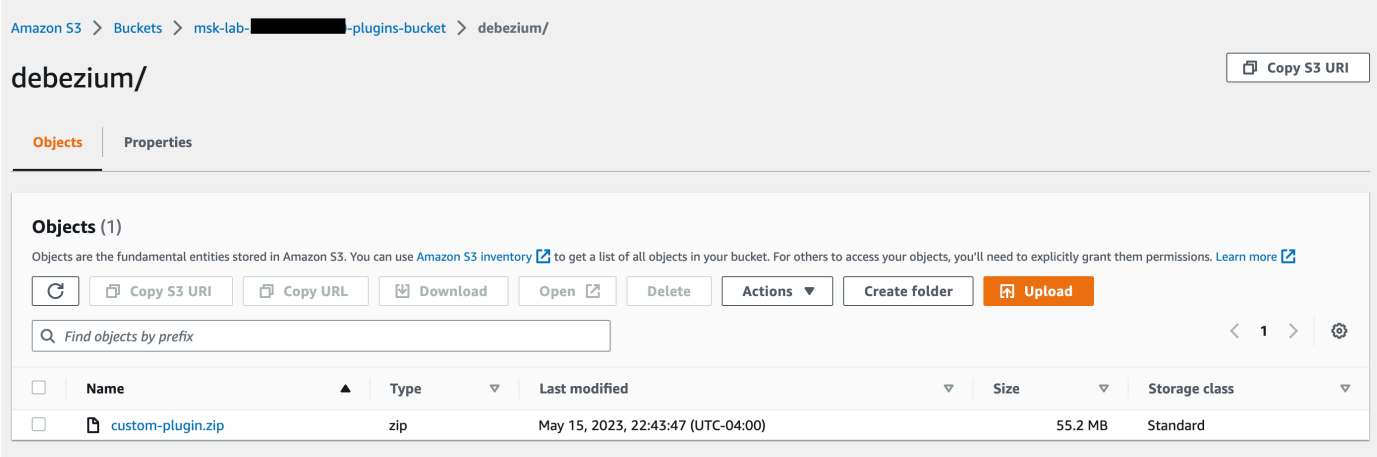
4. 将上述步骤中的 MSK 配置提供程序和自定义连接器的内容压缩到名为 `custom-plugin.zip` 的单个文件中。

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

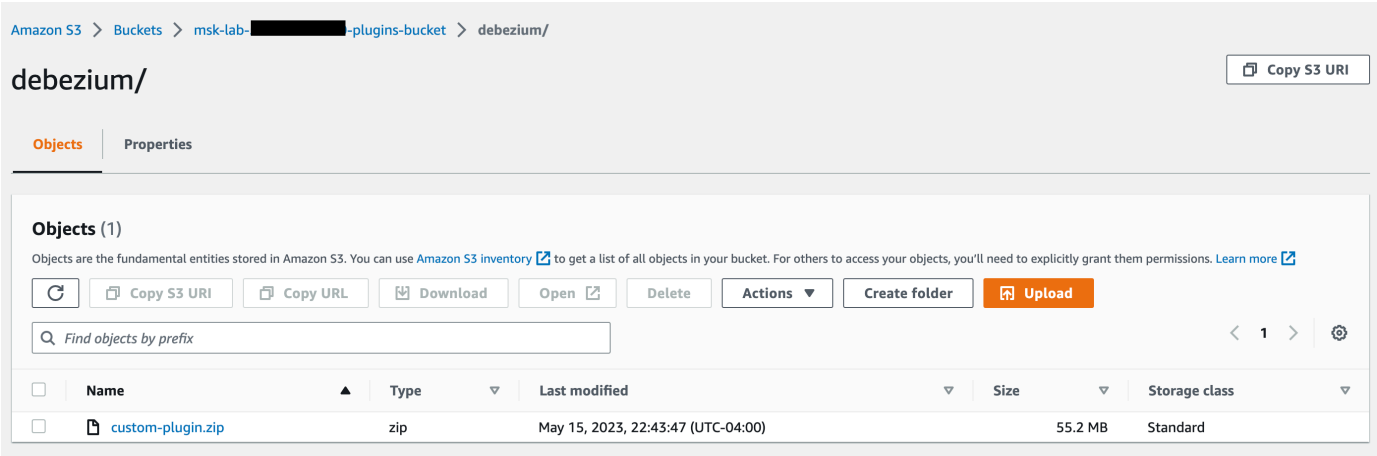
5. 将文件上传到 S3 以供日后参考。

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

6. 在 Amazon MSK 控制台的 MSK Connect 部分下，选择自定义插件，然后选择创建自定义插件并浏览 s3:<***S3_URI_BUCKET_LOCATION***> S3 存储桶，选择刚刚上传的自定义插件 ZIP 文件。



7. 对于插件名称，输入 **debezium-custom-plugin**。或者，输入描述并选择创建自定义插件。



步骤 2：为不同的提供程序配置参数和权限

您可以在以下三个服务中配置参数值：

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

选择以下选项卡之一，获取有关为该服务设置参数和相关权限的说明。

Configure in Secrets Manager

在 Secrets Manager 中配置参数值

1. 打开 Secrets Manager 控制台。
2. 创建新密钥来存储凭证或密钥。有关说明，请参阅《AWS Secrets Manager 用户指南》中的创建 AWS Secrets Manager 密钥。
3. 复制密钥的 ARN。
4. 将以下示例策略中的 Secrets Manager 权限添加到您的服务执行角色。将 *<arn:aws:secretsmanager:us-east-1:123456789000:secret:MySecret-1234>* 替换为密钥的 ARN。
5. 添加工作程序配置和连接器说明。

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

6. 要使用 Secrets Manager 配置提供程序，请在步骤 3 中将以下几行代码复制到工作程序配置文本框中：

```
# 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. 对于 Secrets Manager 配置提供程序，请在步骤 4 中复制连接器配置的以下几行代码。

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

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

您也可以将上述步骤与更多配置提供程序一起使用。

Configure in Systems Manager Parameter Store

在 Systems Manager Parameter Store 中配置参数值

1. 打开 Systems Manager 控制台。
2. 在导航窗格中，选择 Parameter Store。
3. 创建要存储在 Systems Manager 中的新参数。有关说明，请参阅《AWS Systems Manager 用户指南》中的创建 Systems Manager 参数（控制台）。
4. 复制参数的 ARN。
5. 将以下示例策略中的 Systems Manager 权限添加到您的服务执行角色。将 *<arn:aws:ssm:us-east-1:123456789000:parameter/MyParameterName>* 替换为参数的 ARN。

```
{
        "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. 要使用 Parameter Store 配置提供程序，请在步骤 3 中将以下几行代码复制到工作程序配置文本框中：

```
# 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. 对于 Parameter Store 配置提供程序，请在步骤 5 中复制连接器配置的以下几行代码。

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

您也可以将上述两个步骤与更多配置提供程序捆绑使用。

Configure in Amazon S3

在 Amazon S3 中配置对象/文件

1. 打开 Amazon S3 控制台。
2. 将对象上传到 S3 中的存储桶。有关说明，请参阅上传对象。
3. 复制对象的 ARN。

4. 将以下示例策略中的 Amazon S3 对象读取权限添加到您的服务执行角色。将 *<arn:aws:s3:::MY_S3_BUCKET/path/to/custom-plugin.zip>* 替换为对象的 ARN。

```
{
    "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. 要使用 Amazon S3 配置提供程序，请在步骤 3 中将以下几行代码复制到工作程序配置文本框中：

```
# define name of config provider:

config.providers = s3import
# provide implementation classes for S3:

config.providers.s3import.class =
 com.amazonaws.kafka.config.providers.S3ImportConfigProvider
```

6. 对于 Amazon S3 配置提供程序，请在步骤 4 中将以下几行代码复制到连接器配置。

```
#Example implementation for S3 object

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

您也可以将上述两个步骤与更多配置提供程序捆绑使用。

步骤 3：使用与配置提供程序相关的信息创建自定义工作程序配置

1. 在 Amazon MSK Connect 部分下选择工作程序配置。

2. 选择创建工作程序配置。
3. 在“工作程序配置名称”文本框中输入 SourceDebeziumCustomConfig。“描述”是选填项。
4. 根据所需的提供程序复制相关的配置代码，然后将其粘贴到工作程序配置文本框中。
5. 以下是所有三个提供程序的工作程序配置示例：

```
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. 单击“创建工作程序配置”。

步骤 4：创建连接器

1. 按照创建新连接器中的说明，创建新连接器。
2. 选择您在 ??? 中上传到 S3 存储桶中的 custom-plugin.zip 文件作为自定义插件的来源。
3. 根据所需的提供程序复制相关的配置代码，然后将其粘贴到“工作程序配置”字段中。
4. 以下是所有三个提供程序的连接器配置示例：

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

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

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

#Example implementation for Amazon S3 file/object
database.ssl.truststore.location = ${s3import:us-west-2:my_cert_bucket/path/to/
trustore_unique_filename.jks}
```

5. 选择使用自定义配置，然后从工作程序配置下拉列表中选择 SourceDebeziumCustomConfig。
6. 按照创建连接器中说明的其余步骤进行操作。

注意事项

将 MSK 配置提供程序与 Amazon MSK Connect 配合使用时，请考虑以下事项：

- 向 IAM 服务执行角色分配使用配置提供程序时的适当权限。
- 在工作程序配置中定义配置提供程序及其在连接器配置中的实现。
- 如果插件未将敏感配置值定义为秘密，则这些值可能会出现在连接器日志中。Kafka Connect 对未定义的配置值的处理方式与任何其他明文值相同。要了解更多信息，请参阅 防止连接器日志中出现秘密。
- 默认情况下，当连接器使用配置提供程序时，MSK Connect 会经常重新启动该连接器。要关闭此重启行为，可以在连接器配置中将 `config.action.reload` 值设置为 `none`。

MSK Connect 的 IAM 角色和策略

主题

- 服务执行角色
- MSK Connect 的 IAM policy 示例
- 跨服务混淆代理问题防范
- AWS MSK Connect 的托管策略
- 使用 MSK Connect 的服务相关角色

服务执行角色

> **Note**
>
> Amazon MSK Connect 不支持使用服务相关角色作为服务执行角色。您必须创建单独的服务执行角色。有关如何创建自定义 IAM 角色的说明，请参阅 I A M 用户指南中的创建角色以向 AWS 服务委派权限。

使用 MSK Connect 创建连接器时，您需要指定要与之一起使用的 AWS Identity and Access Management （ IAM ）角色。您的服务执行角色必须具有以下信任策略，以便 MSK Connect 可以代入该角色。有关此策略中条件上下文键的说明，请参阅 the section called “跨服务混淆代理问题防范”。

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

如果您想要与连接器一起使用的 Amazon MSK 集群使用 IAM 身份验证，则必须向连接器的服务执行角色添加以下权限策略。有关如何查找集群的 UUID 以及如何构造主题 ARN 的信息，请参阅 the section called “资源”。

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

```
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:DescribeCluster"
            ],
            "Resource": [
                "cluster-arn"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:ReadData",
                "kafka-cluster:DescribeTopic"
            ],
            "Resource": [
                "ARN of the topic that you want a sink connector to read from"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:WriteData",
                "kafka-cluster:DescribeTopic"
            ],
            "Resource": [
                "ARN of the topic that you want a source connector to write to"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:CreateTopic",
                "kafka-cluster:WriteData",
                "kafka-cluster:ReadData",
                "kafka-cluster:DescribeTopic"
            ],
            "Resource": [
                "arn:aws:kafka:region:account-id:topic/cluster-name/cluster-uuid/
__amazon_msk_connect_*"
            ]
        },
        {
```

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

根据连接器的类型，您可能还需要为服务执行角色附加允许其访问 AWS 资源的权限策略。例如，如果您的连接器需要向 S3 存储桶发送数据，则服务执行角色必须具有授予写入该存储桶之权限的权限策略。出于测试目的，您可以使用其中一个预构建 IAM policy 来授予完全访问权限，例如 `arn:aws:iam::aws:policy/AmazonS3FullAccess`。但是，出于安全考虑，我们建议您使用最严格的策略，允许您的连接器从 AWS 源读取数据或写入 AWS 接收器。

MSK Connect 的 IAM policy 示例

要向非管理员用户授予对所有 MSK Connect 功能的完全访问权限，请将如下策略附加到该用户的 IAM 角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafkaconnect:*",
                "ec2:CreateNetworkInterface",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:DescribeSecurityGroups",
                "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries",
                "logs:PutResourcePolicy",
```

```
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
kafkaconnect.amazonaws.com/AWSServiceRoleForKafkaConnect*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "kafkaconnect.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
kafkaconnect.amazonaws.com/AWSServiceRoleForKafkaConnect*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
delivery.logs.amazonaws.com/AWSServiceRoleForLogDelivery*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "delivery.logs.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutBucketPolicy",
                "s3:GetBucketPolicy"
            ],
            "Resource": "ARN of the Amazon S3 bucket to which you want MSK Connect to
 deliver logs"
```

```
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "ARN of the service execution role"
        },
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "ARN of the Amazon S3 object that corresponds to the custom
 plugin that you want to use for creating connectors"
        },
        {
            "Effect": "Allow",
            "Action": "firehose:TagDeliveryStream",
            "Resource": "ARN of the Firehose delivery stream to which you want MSK
 Connect to deliver logs"
        }
    ]
}
```

跨服务混淆代理问题防范

混淆代理问题是一个安全性问题，即不具有操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。在 AWS 中，跨服务模拟可能会导致混淆代理问题。一个服务（呼叫服务）调用另一项服务（所谓的服务)时，可能会发生跨服务模拟。可以操纵调用服务，使用其权限以在其他情况下该服务不应有访问权限的方式对另一个客户的资源进行操作。为了防止这种情况，AWS 提供可帮助您保护所有服务的服务委托人数据的工具，这些服务委托人有权限访问账户中的资源。

我们建议在资源策略中使用 aws:SourceArn 和 aws:SourceAccount 全局条件上下文键，以限制 MSK Connect 为其他服务提供的资源访问权限。如果 aws:SourceArn 值不包含账户 ID，例如 Amazon S3 存储桶 ARN 不包含账户 ID，您必须使用两个全局条件上下文键来限制权限。如果同时使用全局条件上下文密钥和包含账户 ID 的 aws:SourceArn 值，则 aws:SourceAccount 值和 aws:SourceArn 值中的账户在同一策略语句中使用时，必须使用相同的账户 ID。如果您只希望将一个资源与跨服务访问相关联，请使用 aws:SourceArn。如果您想允许该账户中的任何资源与跨服务使用操作相关联，请使用 aws:SourceAccount。

对于 MSK Connect，aws:SourceArn 的值必须是 MSK 连接器。

防范混淆代理问题最有效的方法是使用 aws:SourceArn 全局条件上下文键和资源的完整 ARN。如果不知道资源的完整 ARN，或者正在指定多个资源，请针对 ARN 未知部分使用带有

通配符 (*) 的 aws:SourceArn 全局上下文条件键。例如，*arn:aws:kafkaconnect:us-east-1:123456789012:connector/** 表示属于美国东部（弗吉尼亚州北部）区域内 ID 为 123456789012 的账户的所有连接器。

以下示例演示了如何使用 MSK Connect 中的 aws:SourceArn 和 aws:SourceAccount 全局条件上下文键来防范混淆代理问题。将 *Account-ID* 和 *MSK-Connector-ARN* 替换为您的信息。

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

AWS MSK Connect 的托管策略

AWS 托管策略是由创建和管理的独立策略 AWS。 AWS 托管策略旨在为许多常见用例提供权限，以便您可以开始为用户、组和角色分配权限。

请记住， AWS 托管策略可能不会为您的特定用例授予最低权限权限，因为它们可供所有 AWS 客户使用。我们建议通过定义特定于使用场景的客户管理型策略来进一步减少权限。

您无法更改 AWS 托管策略中定义的权限。如果 AWS 更新 AWS 托管策略中定义的权限，则更新会影响该策略所关联的所有委托人身份（用户、组和角色）。 AWS 最有可能在启动新的 API 或现有服务可以使用新 AWS 服务 的 API 操作时更新 AWS 托管策略。

有关更多信息，请参阅《IAM 用户指南》中的 AWS 托管式策略。

AWS 托管策略：AmazonMSK ConnectReadOnlyAccess

此策略向用户授予列出和描述 MSK Connect 资源所需的权限。

您可以将 AmazonMSKConnectReadOnlyAccess 策略附加到 IAM 身份。

```
{
    "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 托管策略： KafkaConnectServiceRolePolicy

此策略向 MSK Connect 服务授予创建和管理带有 `AmazonMSKConnectManaged:true` 标签的网络接口所需的权限。这些网络接口允许 MSK Connect 通过网络访问 Amazon VPC 中的资源，例如 Apache Kafka 集群、源或接收器。

您无法附加 KafkaConnectServiceRolePolicy 到您的 IAM 实体。此策略附加到服务相关角色，允许 MSK Connect 代表您执行操作。

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "ec2:CreateNetworkInterface"
   ],
   "Resource": "arn:aws:ec2:*:*:network-interface/*",
   "Condition": {
    "StringEquals": {
     "aws:RequestTag/AmazonMSKConnectManaged": "true"
    },
    "ForAllValues:StringEquals": {
     "aws:TagKeys": "AmazonMSKConnectManaged"
    }
   }
  },
  {
   "Effect": "Allow",
   "Action": [
    "ec2:CreateNetworkInterface"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:subnet/*",
    "arn:aws:ec2:*:*:security-group/*"
   ]
  },
  {
   "Effect": "Allow",
```

```
    "Action": [
     "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:*:*:network-interface/*",
    "Condition": {
     "StringEquals": {
      "ec2:CreateAction": "CreateNetworkInterface"
     }
    }
   },
   {
    "Effect": "Allow",
    "Action": [
     "ec2:DescribeNetworkInterfaces",
     "ec2:CreateNetworkInterfacePermission",
     "ec2:AttachNetworkInterface",
     "ec2:DetachNetworkInterface",
     "ec2:DeleteNetworkInterface"
    ],
    "Resource": "arn:aws:ec2:*:*:network-interface/*",
    "Condition": {
     "StringEquals": {
      "ec2:ResourceTag/AmazonMSKConnectManaged": "true"
     }
    }
   }
  ]
}
```

MSK Connect 对 AWS 托管策略的更新

查看自该服务开始跟踪这些更改以来 MSK Connect AWS 托管策略更新的详细信息。

更改	描述	日期
MSK Connect 更新了只读策略	MSK Connect 更新了 AmazonMSK ConnectReadOnlyAccess 政策，取消了对上架操作的限制。	2021 年 10 月 13 日

更改	描述	日期
MSK Connect 开启了跟踪更改	MSK Connect 开始跟踪其 AWS 托管策略的更改。	2021 年 9 月 14 日

使用 MSK Connect 的服务相关角色

Amazon MSK Connect 使用 AWS Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种独特类型的 IAM 角色，它直接链接到 MSK Connect。服务相关角色由 MSK Connect 预定义，包括该服务代表您调用其他 AWS 服务所需的所有权限。

服务相关角色可让您更轻松地设置 MSK Connect，因为您不必手动添加所需权限。MSK Connect 定义其服务相关角色的权限，除非另外定义，否则只有 MSK Connect 可以代入其角色。定义的权限包括信任策略和权限策略，而且权限策略不能附加到任何其它 IAM 实体。

有关支持服务相关角色的其他服务的信息，请参阅与 IAM 结合使用的AWS 服务，并查找服务相关角色列中显示为是的服务。选择是，可转到查看该服务的服务相关角色文档的链接。

MSK Connect 的服务相关角色权限

MSK Connect 使用名为 AWSServiceRoleForKafkaConnect— 允许亚马逊 MSK Connect 代表您访问亚马逊资源的服务相关角色。

AWSServiceRoleForKafkaConnect 服务相关角色信任`kafkaconnect.amazonaws.com`服务来代替该角色。

有关该角色使用的权限策略的信息，请参阅 the section called “KafkaConnectServiceRolePolicy”。

您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的服务相关角色权限。

创建 MSK Connect 的服务相关角色

您无需手动创建服务相关角色。当您在 AWS Management Console、或 AWS API 中创建连接器时 AWS CLI，MSK Connect 会为您创建服务相关角色。

如果您删除该服务相关角色，然后需要再次创建，您可以使用相同流程在账户中重新创建此角色。当您创建连接器时，MSK Connect 将再次为您创建服务相关角色。

编辑 MSK Connect 的服务相关角色

MSK Connect 不允许您编辑 AWSServiceRoleForKafkaConnect 服务相关角色。在创建服务相关角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。不过，您可以使用 IAM 编辑角色的说明。有关更多信息，请参阅《IAM 用户指南》中的编辑服务相关角色。

删除 MSK Connect 的服务相关角色

您可以使用 IAM 控制台、 AWS CLI 或 AWS API 手动删除服务相关角色。为执行此操作，您必须先手动删除所有 MSK Connect 连接器，然后才能手动删除该角色。有关更多信息，请参见《IAM 用户指南》中的删除服务相关角色。

MSK Connect 服务相关角色的受支持区域

MSK Connect 支持在该服务可用的所有区域中使用服务相关角色。有关更多信息，请参阅 AWS 区域和端点。

为 Amazon MSK Connect 启用互联网访问

如果您的 Amazon MSK Connect 连接器需要访问互联网，我们建议您使用以下 Amazon Virtual Private Cloud (VPC) 设置来启用该访问权限。

- 使用私有子网配置连接器。
- 在公有子网中为您的 VPC 创建公有 NAT 网关或 NAT 实例。有关更多信息，请参阅《Amazon Virtual Private Cloud用户指南》中的使用 NAT 设备将子网连接到互联网或其他 VPC 页面。
- 允许从私有子网到 NAT 网关或实例的出站流量。

为 Amazon MSK Connect 设置 NAT 网关

以下步骤显示如何设置 NAT 网关，以便为连接器启用互联网访问。在私有子网中创建连接器之前，必须完成这些步骤。

先决条件

确保您已具有以下项目。

- 与您的集群关联的 Amazon Virtual Private Cloud (VPC) 的 ID。例如 vpc-123456ab。

- VPC 中的私有子网 ID。例如 subnet-a1b2c3de、subnet-f4g5h6ij 等。您必须使用私有子网配置连接器。

为连接器启用互联网访问

1. 打开 Amazon Virtual Private Cloud 控制台，网址为 https://console.aws.amazon.com/vpc/。
2. 使用描述性名称为您的 NAT 网关创建一个公有子网，并记下子网 ID。有关详细说明，请参阅在 VPC 中创建子网。
3. 创建互联网网关以便您的 VPC 可以与互联网通信，并记下网关 ID。将互联网网关附加到 VPC。有关说明，请参阅创建并附加互联网网关。
4. 预置公有 NAT 网关，以便私有子网中的主机可以访问您的公有子网。创建 NAT 网关时，请选择之前创建的公有子网。有关说明，请参阅创建 NAT 网关。
5. 配置路由表。您总共必须有两个路由表才能完成此设置。您应该已经有一个与您的 VPC 同时自动创建的主路由表。在此步骤中，您需为公有子网创建额外的路由表。

   a. 使用以下设置修改 VPC 的主路由表，以便私有子网将流量路由到您的 NAT 网关。有关说明，请参阅《Amazon Virtual Private Cloud用户指南》中的使用路由表。

      私有 MSKC 路由表

属性	值
名称标签	建议您为该路由表指定一个描述性的名称标签，以帮助您识别它。例如私有 MSKC。
关联的子网	您的私有子网
为 MSK Connect 启用互联网访问的路由	• 目的地：0.0.0.0/0 • 目标：您的 NAT 网关 ID。例如 nat-12a345bc6789efg1h。
内部流量的本地路由	• 目的地：10.0.0.0/16。此值可能会有所不同，具体取决于您 VPC 的 CIDR 块。 • 目标：本地

   b. 按照创建自定义路由表中的说明，为公有子网创建路由表。创建表时，在名称标签字段中输入描述性名称，以帮助您识别该表与哪个子网关联。例如公有 MSKC。

   c. 使用以下设置配置您的公有 MSKC 路由表。

属性	值
名称标签	公有 MSKC 或您选择的其他描述性名称
关联的子网	带有 NAT 网关的公有子网
为 MSK Connect 启用互联网访问的路由	• 目的地：0.0.0.0/0 • 目标：您的互联网网关 ID。例如 igw-1a234bc5。
内部流量的本地路由	• 目的地：10.0.0.0/16。此值可能会有所不同，具体取决于您 VPC 的 CIDR 块。 • 目标：本地

私有 DNS 主机名

借助 MSK Connect 中的私有 DNS 主机名支持，您可以配置连接器以参考公有或私有域名。支持取决于 VPC DHCP 选项集中指定的 DNS 服务器。

DHCP 选项集是一组网络配置，可供 VPC 中的 EC2 实例用于通过 VPC 网络进行通信。每个 VPC 都有一个默认 DHCP 选项集，但如果您希望 VPC 中的实例使用不同的 DNS 服务器来进行域名解析，而不使用 Amazon 提供的 DNS 服务器，您也可以创建自定义 DHCP 选项集。参阅 Amazon VPC 中的 DHCP 选项集。

连接器使用服务 VPC DNS 解析器从客户连接器进行 DNS 查询后，才能在 MSK Connect 中包含私有 DNS 解析能力/功能。连接器未使用客户 VPC DHCP 选项集中定义的 DNS 服务器进行 DNS 解析。

连接器只能参考在客户连接器配置或插件中可公开解析的主机名。它们无法解析在私有托管区中定义的私有主机名，也无法在其他客户网络中使用 DNS 服务器。

如果没有私有 DNS，那些选择让自己的数据库、数据仓库和系统（例如自己 VPC 中的 Secrets Manager）无法访问互联网的客户就无法使用 MSK 连接器。客户经常使用私有 DNS 主机名来遵循企业安全状况要求。

主题

- 配置连接器的 VPC DHCP 选项集
- VPC 中的 DNS 属性

配置连接器的 VPC DHCP 选项集

创建连接器时，连接器会自动使用在其 VPC DHCP 选项集中定义的 DNS 服务器。在创建连接器之前，请确保已为连接器的 DNS 主机名解析要求配置 VPC DHCP 选项集。

在 MSK Connect 中提供私有 DNS 主机名功能之前创建的连接器将继续使用之前的 DNS 解析配置，无需进行任何修改。

如果您只需要在连接器中进行可公开解析的 DNS 主机名解析，为了更容易设置，建议您在创建连接器时使用账户的默认 VPC。有关 Amazon 提供的 DNS 服务器或 Amazon Route 53 Resolver 的更多信息，请参阅《Amazon VPC 用户指南》中的 Amazon DNS 服务器。

如果您需要解析私有 DNS 主机名，请确保在创建连接器过程中传递的 VPC 的 DHCP 选项集已正确配置。有关更多信息，请参阅《Amazon VPC 用户指南》中的使用 DHCP 选项集。

在配置私有 DNS 主机名解析的 DHCP 选项集时，请确保连接器可以访问您在 DHCP 选项集中配置的自定义 DNS 服务器。否则，连接器将创建失败。

自定义 VPC DHCP 选项集后，随后在该 VPC 中创建的连接器将使用您在选项集中指定的 DNS 服务器。如果在创建连接器后更改选项集，则该连接器将在几分钟内采用新选项集中的设置。

VPC 中的 DNS 属性

确保已按照《Amazon VPC 用户指南》的您 VPC 中的 DNS 属性和 DNS 主机名中所述正确配置 VPC DNS 属性。

有关使用入站和出站解析器端点将其他网络连接到 VPC 以使用您的连接器的信息，请参阅《Amazon Route 53 开发人员指南》中的解析 VPC 与您的网络之间的 DNS 查询。

故障处理

本节介绍与 DNS 解析相关的可能的连接器创建失败以及解决问题的建议操作。

失败	建议采取的措施
如果 DNS 解析查询失败，或者无法从连接器访问 DNS 服务器，则连接器创建失败。	如果您已为连接器配置 CloudWatch 日志，则可以在这些日志中看到因 DNS 解析查询失败而导致的连接器创建失败。

失败	建议采取的措施
	检查 DNS 服务器配置，并确保从连接器到 DNS 服务器的网络连接可用。
如果在连接器运行时更改 VPC DHCP 选项集中的 DNS 服务器配置，则来自连接器的 DNS 解析查询可能会失败。如果 DNS 解析失败，某些连接器任务可能会进入失败状态。	如果您已为连接器配置 CloudWatch 日志，则可以在这些日志中看到因 DNS 解析查询失败而导致的连接器创建失败。 失败的任务应自动重启以使连接器恢复正常。如果没有发生这种情况，您可以联系支持人员为其连接器重启失败的任务，也可以重新创建连接器。

为 MSK Connect 进行日志记录

MSK Connect 可以写入可用于调试连接器的日志事件。创建连接器时，您可以指定零个或多个以下日志目标：

- Amazon CloudWatch 日志：您可以指定希望 MSK Connect 将连接器的日志事件发送到哪个日志组。有关如何创建日志组的信息，请参阅《日志用户指南》中的创建CloudWatch 日志组。
- Amazon S3：您可以指定希望 MSK Connect 向其发送连接器日志事件的 S3 存储桶。有关如何创建 S3 存储桶的信息，请参阅《Amazon S3 用户指南》中的创建存储桶。
- Amazon Data Firehose：您可以指定希望 MSK Connect 将连接器日志事件发送到的传输流。有关如何创建传输流的信息，请参阅 Firehose 用户指南中的创建 Amazon Data Firehose 传输流。

要了解有关设置日志记录的更多信息，请参阅《Amazon CloudWatch Logs 用户指南》中的启用从某些 AWS 服务进行日志记录。

MSK Connect 会发出以下类型的日志事件：

级别	描述
`INFO`	启动和关闭时感兴趣的运行时系统事件。
`WARN`	不是错误但不希望出现或意外的运行时系统情况。

级别	描述
FATAL	导致过早终止的严重错误。
ERROR	非致命的意外情况和运行时系统错误。

以下是发送到 Log CloudWatch s 的日志事件的示例：

```
[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)
```

防止连接器日志中出现秘密

> **Note**
>
> 如果插件未将敏感配置值定义为秘密，则这些值可能会出现在连接器日志中。Kafka Connect 对未定义的配置值的处理方式与任何其他明文值相同。

如果您的插件将某个属性定义为秘密，则 Kafka Connect 会从连接器日志中编辑该属性的值。例如，以下连接器日志表明，如果插件将 `aws.secret.key` 定义为 PASSWORD 类型，则其值将替换为 **`[hidden]`**。

```
   2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b] [2022-01-11
15:18:55,150] INFO SecretsManagerConfigProviderConfig values:
   2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b] aws.access.key =
my_access_key
   2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b] aws.region = us-east-1
   2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b] aws.secret.key
= [hidden]
   2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b] secret.prefix =
   2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b] secret.ttl.ms = 300000
   2022-01-11T15:18:55.000+00:00    [Worker-05e6586a48b5f331b]
(com.github.jcustenborder.kafka.config.aws.SecretsManagerConfigProviderConfig:361)
```

为防止连接器日志文件中出现秘密，插件开发人员必须使用 Kafka Connect 枚举常量 `ConfigDef.Type.PASSWORD` 来定义敏感属性。当属性为类型 `ConfigDef.Type.PASSWORD` 时，Kafka Connect 会将其值从连接器日志中排除，即使该值以明文形式发送也一样。

监控 MSK Connect

监控对于保持 MSK Connect 和其他 AWS 解决方案的可靠性、可用性和性能十分重要。Amazon CloudWatch 实时监控您的 AWS 资源以及在 AWS 上运行的应用程序。您可以收集和跟踪指标，创建自定义的控制面板，以及设置警报以在指定的指标达到您指定的阈值时通知您或采取措施。例如，您可以让 CloudWatch 跟踪 CPU 使用率或连接器的其他指标，以便在需要时增加其容量。有关更多信息，请参阅 Amazon CloudWatch 用户指南。

下表显示了 MSK Connect 在 `ConnectorName` 维度下发送给 CloudWatch 的指标。MSK Connect 默认会提供这些指标，无需额外费用。CloudWatch 会保留这些指标 15 个月，以便您可以访问历史信息，更好地了解连接器的执行情况。此外，可以设置用于监测特定阈值的警报，并在达到相应阈值时发送通知或执行操作。有关更多信息，请参阅 Amazon CloudWatch 用户指南。

MSK Connect 指标

指标名称	描述
`BytesInPerSec`	连接器接收的总字节数。
`BytesOutPerSec`	连接器传送的总字节数。
`CpuUtilization`	系统和用户的 CPU 消耗百分比。
`ErroredTaskCount`	已出错的任务数量。
`MemoryUtilization`	工作程序实例上总内存的百分比，而不仅仅是当前正在使用的 Java 虚拟机（JVM）堆内存。JVM 通常不会将内存释放回操作系统。因此，JVM 堆大小（MemoryUtilization）通常从最小堆大小开始，逐渐增加到约 80-90% 的稳定最大值。随着连接器实际内存使用量的变化，JVM 堆使用量可能会增加或减少。
`RebalanceCompletedTotal`	此连接器完成的重新平衡总数。

指标名称	描述
RebalanceTimeAvg	连接器在重新平衡上花费的平均时间（以毫秒为单位）。
RebalanceTimeMax	连接器在重新平衡上花费的最长时间（以毫秒为单位）。
RebalanceTimeSinceLast	自此连接器完成最近一次重新平衡以来的时间（以毫秒为单位）。
RunningTaskCount	连接器中正在运行的任务数量。
SinkRecordReadRate	平均每秒从 Apache Kafka 或 Amazon MSK 集群读取的记录数量。
SinkRecordSendRate	平均每秒从转换中输出并发送到目标的记录数量。此数量不包含筛选后的记录。
SourceRecordPollRate	平均每秒生成或轮询的记录数量。
SourceRecordWriteRate	平均每秒从转换中输出并写入 Apache Kafka 或 Amazon MSK 集群的记录数量。
TaskStartupAttemptsTotal	连接器已尝试的任务启动总数。您可以使用此指标来识别任务启动尝试中的异常情况。
TaskStartupSuccessPercentage	连接器成功启动任务的平均百分比。您可以使用此指标来识别任务启动尝试中的异常情况。
WorkerCount	在连接器中运行的工作程序数量。

示例

本节包含一些示例，可帮助您设置 Amazon MSK Connect 资源，例如常见的第三方连接器和配置提供程序。

主题

- Amazon S3 接收器连接器

- 带有配置提供程序的 Debezium 源连接器

Amazon S3 接收器连接器

此示例演示了如何使用 Confluent Amazon S3 接收器连接器和 AWS CLI 在 MSK Connect 中创建 Amazon S3 接收器连接器。

1. 复制以下 JSON 并将其粘贴到新文件中。将占位符字符串替换为与 Amazon MSK 集群的引导服务器连接字符串以及集群的子网和安全组 ID 相对应的值。有关如何设置服务执行角色的信息，请参阅 the section called “IAM 角色和策略”。

```
{
    "connectorConfiguration": {
        "connector.class": "io.confluent.connect.s3.S3SinkConnector",
        "s3.region": "us-east-1",
        "format.class": "io.confluent.connect.s3.format.json.JsonFormat",
        "flush.size": "1",
        "schema.compatibility": "NONE",
        "topics": "my-test-topic",
        "tasks.max": "2",
        "partitioner.class":
 "io.confluent.connect.storage.partitioner.DefaultPartitioner",
        "storage.class": "io.confluent.connect.s3.storage.S3Storage",
        "s3.bucket.name": "my-test-bucket"
    },
    "connectorName": "example-S3-sink-connector",
    "kafkaCluster": {
        "apacheKafkaCluster": {
            "bootstrapServers": "<cluster-bootstrap-servers-string>",
            "vpc": {
                "subnets": [
                    "<cluster-subnet-1>",
                    "<cluster-subnet-2>",
                    "<cluster-subnet-3>"
                ],
                "securityGroups": ["<cluster-security-group-id>"]
            }
        }
    },
    "capacity": {
        "provisionedCapacity": {
            "mcuCount": 2,
```

```
            "workerCount": 4
        }
    },
    "kafkaConnectVersion": "2.7.1",
    "serviceExecutionRoleArn": "<arn-of-a-role-that-msk-connect-can-assume>",
    "plugins": [
        {
            "customPlugin": {
                "customPluginArn": "<arn-of-custom-plugin-that-contains-connector-
code>",
                "revision": 1
            }
        }
    ],
    "kafkaClusterEncryptionInTransit": {"encryptionType": "PLAINTEXT"},
    "kafkaClusterClientAuthentication": {"authenticationType": "NONE"}
}
```

2. 在上一步中保存 JSON 文件的文件夹中运行以下 AWS CLI 命令。

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

以下是您在成功运行命令后获得的输出示例。

```
{
    "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"
}
```

带有配置提供程序的 Debezium 源连接器

此示例演示了如何将 Debezium MySQL 连接器插件与兼容 MySQL 的 Amazon Aurora 数据库一起用作来源。在此示例中，我们还设置了开源 AWS Secrets Manager 配置提供程序来对 AWS Secrets Manager 中的数据库凭证进行外部化。要了解有关配置提供程序的更多信息，请参阅 使用配置提供程序将敏感信息外部化。

> **Important**
>
> Debezium MySQL 连接器插件仅支持一项任务，不使用 Amazon MSK Connect 的自动扩缩容量模式。您应该改为使用预置容量模式，并在连接器配置中将 `workerCount` 设置为 1。要了解有关 MSK Connect 容量模式的更多信息，请参阅 连接器容量。

开始前的准备工作

您的连接器必须能够访问互联网，这样才能与 Amazon Virtual Private Cloud 之外的服务（例如 AWS Secrets Manager）进行交互。本节中的步骤可帮助您完成以下任务以启用互联网访问。

- 设置托管 NAT 网关并将流量路由到 VPC 中互联网网关的公有子网。
- 创建将私有子网流量定向到 NAT 网关的默认路由。

有关更多信息，请参阅 为 Amazon MSK Connect 启用互联网访问。

先决条件

在启用互联网访问之前，您需要以下项目：

- 与集群关联的 Amazon Virtual Private Cloud（VPC）ID。例如 vpc-123456ab。
- VPC 中的私有子网 ID。例如 subnet-a1b2c3de、subnet-f4g5h6ij 等。您必须使用私有子网配置连接器。

为连接器启用互联网访问

1. 打开 Amazon Virtual Private Cloud 控制台 (https://console.aws.amazon.com/vpc/)。
2. 使用描述性名称为您的 NAT 网关创建一个公有子网，并记下子网 ID。有关详细说明，请参阅在 VPC 中创建子网。
3. 创建互联网网关以便您的 VPC 可以与互联网通信，并记下网关 ID。将互联网网关附加到 VPC。有关说明，请参阅创建并附加互联网网关。
4. 预置公有 NAT 网关，以便私有子网中的主机可以访问您的公有子网。创建 NAT 网关时，请选择之前创建的公有子网。有关说明，请参阅创建 NAT 网关。
5. 配置路由表。您总共必须有两个路由表才能完成此设置。您应该已经有一个与您的 VPC 同时自动创建的主路由表。在此步骤中，您需为公有子网创建额外的路由表。

a. 使用以下设置修改 VPC 的主路由表，以便私有子网将流量路由到您的 NAT 网关。有关说明，请参阅《Amazon Virtual Private Cloud 用户指南》中的使用路由表。

   私有 MSKC 路由表

属性	Value
名称标签	建议您为该路由表指定一个描述性的名称标签，以帮助您识别它。例如私有 MSKC。
关联的子网	您的私有子网
为 MSK Connect 启用互联网访问的路由	• 目的地：0.0.0.0/0 • 目标：您的 NAT 网关 ID。例如 nat-12a345bc6789efg1h。
内部流量的本地路由	• 目的地：10.0.0.0/16。此值可能会有所不同，具体取决于您 VPC 的 CIDR 块。 • 目标：本地

b. 按照创建自定义路由表中的说明，为公有子网创建路由表。创建表时，在名称标签字段中输入描述性名称，以帮助您识别该表与哪个子网关联。例如公有 MSKC。

c. 使用以下设置配置您的公有 MSKC 路由表。

属性	Value
名称标签	公有 MSKC 或您选择的其他描述性名称
关联的子网	带有 NAT 网关的公有子网
为 MSK Connect 启用互联网访问的路由	• 目的地：0.0.0.0/0 • 目标：您的互联网网关 ID。例如 igw-1a234bc5。
内部流量的本地路由	• 目的地：10.0.0.0/16。此值可能会有所不同，具体取决于您 VPC 的 CIDR 块。 • 目标：本地

现在，您已经为 Amazon MSK Connect 启用互联网访问，可以创建连接器了。

创建 Debezium 源连接器

1. 创建自定义插件

   a. 从 Debezium 网站下载 MySQL 连接器插件的最新稳定发行版。记下您下载的 Debezium 发行版（版本 2.x 或较旧的 1.x 系列）。在此程序的后面部分，您需根据您的 Debezium 版本创建连接器。

   b. 下载并解压缩 AWS Secrets Manager 配置提供程序。

   c. 将以下档案文件放在同一个目录中：

   - debezium-connector-mysql 文件夹
   - jcusten-border-kafka-config-provider-aws-0.1.1 文件夹

   d. 将您在上一步中创建的目录压缩为 ZIP 文件，然后将该 ZIP 文件上传到 S3 存储桶。有关说明，请参阅《Amazon S3 用户指南》中的上传对象。

   e. 复制以下 JSON 并将其粘贴到文件中。例如，debezium-source-custom-plugin.json。将 *<example-custom-plugin-name>* 替换为您想要的插件名称，将 *<arn-of-your-s3-bucket>* 替换为上传 ZIP 文件的 S3 存储桶的 ARN，以及将 *<file-key-of-ZIP-object>* 替换为上传到 S3 的 ZIP 对象的文件密钥。

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

   f. 从保存 JSON 文件的文件夹中运行以下 AWS CLI 命令来创建插件。

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

   您应该可以看到类似于以下示例的输出内容。

```
{
    "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. 运行以下命令以检查插件状态。状态应从 CREATING 更改为 ACTIVE。将 ARN 占位符替换为您在上一条命令的输出中获得的 ARN。

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

2. 为您的数据库凭证配置 AWS Secrets Manager 并创建密钥

a. 在 https://console.aws.amazon.com/secretsmanager/ 打开 Secrets Manager 控制台

b. 创建新密钥来存储您的数据库登录凭证。有关说明，请参阅《AWS Secrets Manager 用户指南》中的创建密钥。

c. 复制密钥的 ARN。

d. 将以下示例策略中的 Secrets Manager 权限添加到您的 服务执行角色。将 *<arn:aws:secretsmanager:us-east-1:123456789000:secret:MySecret-1234>* 替换为密钥的 ARN。

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

```
}
```

有关如何添加 IAM 权限的说明，请参阅《IAM 用户指南》中的添加和删除 IAM 身份权限。

3. 使用与配置提供程序有关的信息创建自定义工作程序配置

   a. 将以下工作程序配置属性复制到文件中，将占位符字符串替换为与您的场景对应的值。要了解 AWS Secrets Manager 配置提供程序的配置属性，请参阅插件文档中的 SecretsManagerConfigProvider。

```
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. 运行以下 AWS CLI 命令以创建自定义工作程序配置。

   替换以下值：

   - *<my-worker-config-name>* – 自定义工作程序配置的描述性名称
   - *<encoded-properties-file-content-string>* – 您在上一步中复制的明文属性的 base64 编码版本

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

4. 创建连接器

   a. 复制与 Debezium 版本（2.x 或 1.x）相对应的以下 JSON，并将其粘贴到新文件中。将 *<placeholder>* 字符串替换为与您的场景对应的值。有关如何设置服务执行角色的信息，请参阅 the section called “IAM 角色和策略”。

   请注意，该配置使用诸如 `${secretManager:MySecret-1234:dbusername}` 之类的变量而不是明文来指定数据库凭证。将 *MySecret-1234* 替换为密钥名称，然后加入您想要检索的密钥名称。您还必须将 *<arn-of-config-provider-worker-configuration>* 替换为自定义工作程序配置的 ARN。

Debezium 2.x

对于 Debezium 2.x 版本，请复制以下 JSON 并将其粘贴到新文件中。将 *<placeholder>* 字符串替换为与您的场景对应的值。

```
{
 "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

对于 Debezium 1.x 版本，请复制以下 JSON 并将其粘贴到新文件中。将 *<placeholder>* 字符串替换为与您的场景对应的值。

```
{
  "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. 在上一步中保存 JSON 文件的文件夹中运行以下 AWS CLI 命令。

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

以下是您在成功运行命令后获得的输出示例。

```
{
    "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"
}
```

有关包含详细步骤的 Debezium 连接器示例，请参阅 Introducing Amazon MSK Connect - Stream Data to and from Your Apache Kafka Clusters Using Managed Connectors。

最佳实践

使用此信息作为参考可以快速找到使用 Amazon MSK Connect 最大程度地提高性能的建议。

通过连接器连接

以下最佳实践可以提高您与 Amazon MSK Connect 的连接性能。

请勿使 Amazon VPC 对等连接或中转网关的 IP 重叠

如果您通过 Amazon MSK Connect 使用 Amazon VPC 对等连接或中转网关，请勿将连接器配置为使用 CIDR 范围内的 IP 访问对等 VPC 资源：

- “10.99.0.0/16”
- “192.168.0.0/16”
- “172.21.0.0/16”

亚马逊 MSK Connect 迁移指南

本节介绍如何将你的 Apache Kafka 连接器应用程序迁移到适用于 Apache Kafka 的亚马逊托管流媒体 Kafka Connect（亚马逊 MSK Connect）。

主题

使用 Amazon MSK Connect 的好处

Apache Kafka 是用于摄取和处理实时数据流的最广泛采用的开源流媒体平台之一。借助 Apache Kafka，您可以分离和独立扩展数据生成和数据消耗应用程序。

Kafka Connect 是使用 Apache Kafka 构建和运行流媒体应用程序的重要组成部分。Kafka Connect 提供了一种在 Kafka 和外部系统之间移动数据的标准化方式。Kafka Connect 具有高度可扩展性，可以处理大量数据 Kafka Connect 提供了一组强大的 API 操作和工具，用于配置、部署和监控在 Kafka 主题和外部系统之间移动数据的连接器。您可以使用这些工具自定义和扩展 Kafka Connect 的功能，以满足您的流媒体应用程序的特定需求。

当你自己操作 Apache Kafka Connect 集群时，或者当你尝试将开源 Apache Kafka Connect 应用程序迁移到时，你可能会遇到挑战。AWS这些挑战包括设置基础架构和部署应用程序所需的时间、设置自我管理的 Apache Kafka Connect 集群时的工程障碍以及管理操作开销。

为了应对这些挑战，我们建议使用适用于 Apache Kafka 的亚马逊托管流媒体 Kafka Connect（亚马逊 MSK Connect）将你的开源 Apache Kafka Connect 应用程序迁移到。AWSAmazon MSK Connect 简化了使用 Kafka Connect 在 Apache Kafka 集群和外部系统（例如数据库、搜索索引和文件系统）之间流入和流出数据。

以下是迁移到 Amazon MSK Connect 的一些好处：

- 消除运营开销 — Amazon MSK Connect 消除了与 Apache Kafka Connect 集群的修补、配置和扩展相关的运营负担。Amazon MSK Connect 会持续监控您的 Connect 集群的运行状况，并自动进行修补和版本升级，而不会对您的工作负载造成任何中断。
- 自动重启 Connect 任务 — Amazon MSK Connect 可以自动恢复失败的任务以减少生产中断。任务失败可能是由临时错误引起的，例如突破 Kafka 的 TCP 连接限制，以及新工作人员加入接收器连接器的使用者组时的任务重新平衡。
- 自动水平和垂直扩展 — Amazon MSK Connect 使连接器应用程序能够自动扩展以支持更高的吞吐量。Amazon MSK Connect 为您管理扩展。您只需要指定 auto Scaling 组中的工作人员数量和利用率阈值即可。您可以使用 Amazon MSK Connect `UpdateConnector` API 操作在 1 到 8 个 vCPU 之间纵向扩展或缩小 vCPU，以支持可变吞吐量。
- 私有网络连接 — Amazon MSK Connect 使用私有 DNS 名称私密连接到源系统AWS PrivateLink 和接收系统。

迁移到亚马逊 MSK Connect

本节简要介绍了 Kafka Connect 和 Amazon MSK Connect 使用的状态管理主题。本节还介绍迁移源连接器和接收器连接器的过程。

主题

Kafka Connect 使用的内部话题

在分布式模式下运行的 Apache Kafka Connect 应用程序使用 Kafka 集群中的内部主题和组成员资格来存储其状态。以下是与 Kafka Connect 应用程序使用的内部主题相对应的配置值：

- 配置主题，通过指定 `config.storage.topic`

  在配置主题中，Kafka Connect 存储了用户启动的所有连接器和任务的配置。每当用户更新连接器的配置或连接器请求重新配置时（例如，连接器检测到它可以启动更多任务），都会向该主题发出一条记录。本主题已启用压缩，因此它始终保留每个实体的最后一个状态。

- 偏移主题，通过指定 `offset.storage.topic`

  在偏移量主题中，Kafka Connect 存储了源连接器的偏移量。与配置主题一样，偏移量主题已启用压缩。本主题仅用于为从外部系统向 Kafka 生成数据的源连接器写入源位置。Sink 连接器从 Kafka 读取数据并发送到外部系统，使用常规 Kafka 使用者组存储其消费者偏移量。

- 状态主题，通过指定 `status.storage.topic`

  在状态主题中，Kafka Connect 存储了连接器和任务的当前状态。本主题用作 REST API 用户查询数据的中心位置。本主题允许用户查询任何 worker，但仍能获取所有正在运行的插件的状态。与配置和偏移主题一样，状态主题也启用了压缩。

除了这些话题外，Kafka Connect 还广泛使用了 Kafka 的群组成员资格 API。这些组以连接器名称命名。例如，对于名为 file-sink 的连接器，该组被命名为。 connect-file-sink组中的每个使用者都为单个任务提供记录。可以使用常规的消费者组工具（例如）来检索这些组及其抵消量。`Kafka-consumer-group.sh`对于每个接收器连接器，Connect 运行时都会运行一个从 Kafka 中提取记录的常规使用者组。

亚马逊 MSK Connect 应用程序的状态管理

默认情况下，Amazon MSK Connect 在 Kafka 集群中为每个 Amazon MSK 连接器创建三个单独的主题，用于存储连接器的配置、偏移量和状态。默认主题名称的结构如下：

- *__msk_connect_configs_ ##### _ ### ID*
- *__msk_connect_status_ ##### _ ### ID*
- *__msk_connect_offsets_ ##### _ #####*

> **Note**
>
> 要提供源连接器之间的偏移连续性，您可以使用自己选择的偏移存储主题，而不是默认的主题。指定偏移存储主题可以帮助您完成创建源连接器之类的任务，该连接器可从上一个连接器的最后一个偏移恢复读取。要指定偏移存储主题，请在创建连接器之前在 Amazon MSK Connect 工作程序配置中为该offset.storage.topic属性提供一个值。

将源连接器迁移到 Amazon MSK Connect

源连接器是将记录从外部系统导入到 Kafka 的 Apache Kafka Connect 应用程序。本节介绍将本地运行的 Apache Kafka Connect 源连接器应用程序迁移到亚马逊 MSK Connect 的过程，或者将运行的自管理 Kafka Connect 集群迁移到AWS亚马逊 MSK Connect。

Kafka Connect 源连接器应用程序将偏移量存储在主题中，该主题以为配置属性设置的值命名。`offset.storage.topic`以下是 JDBC 连接器的偏移消息示例，该连接器正在运行两个任务，这些任务从名`movies`为和的两个不同表导入数据。`shows`最近从表格电影中导入的行的主 ID 为`18343`。从 shows 表中导入的最新行的主 ID 为`732`。

```
["jdbcsource",{"protocol":"1","table":"sample.movies"}] {"incrementing":18343}
["jdbcsource",{"protocol":"1","table":"sample.shows"}] {"incrementing":732}
```

要将源连接器迁移到 Amazon MSK Connect，请执行以下操作：

1. 通过从本地或自行管理的 Kafka Connect 集群中提取连接器库，创建 Amazon MSK Connect 自定义插件。
2. 创建 Amazon MSK Connect 工作程序属性`key.converter`，并将属性`value.converter`、和`offset.storage.topic`设置为为在现有 Kafka Connect 集群中运行的 Kafka 连接器设置的相同值。
3. 通过在现有 Kafka Connect 集群上`PUT /connectors/`*`connector-name`*`/pause`发出请求，暂停现有集群上的连接器应用程序。
4. 确保连接器应用程序的所有任务都已完全停止。您可以通过在现有 Kafka Connect 集群上`GET /connectors/`*`connector-name`*`/status`发出请求或使用为该属性`status.storage.topic`设置的主题名称中的消息来停止任务。
5. 从现有集群获取连接器配置。您可以通过在现有集群上发出`GET /connectors/`*`connector-name`*`/config/`请求或使用为该属性设置的主题名称中的消息来获取连接器配置`config.storage.topic`。

6. 创建一个与现有集群同名的新 Amazon MSK 连接器。使用您在步骤 1 中创建的连接器自定义插件、在步骤 2 中创建的工作器属性和在步骤 5 中提取的连接器配置来创建此连接器。
7. 当 Amazon MSK 连接器状态为时`active`，查看日志以验证连接器是否已开始从源系统导入数据。
8. 通过提出`DELETE /connectors/`*`connector-name`*请求来删除现有集群中的连接器。

将接收器连接器迁移到亚马逊 MSK Connect

接收器连接器是 Apache Kafka Connect 应用程序，用于将数据从 Kafka 导出到外部系统。本节介绍将本地运行的 Apache Kafka Connect 接收器应用程序迁移到亚马逊 MSK Connect 的过程，或者将运行的自管理 Kafka Connect 集群迁移到 AWS Amazon MSK Connect。

Kafka Connect 接收器连接器使用 Kafka 群组成员资格 API，并将偏移存储在与典型使用者应用程序相同的`__consumer_offset`主题中。此行为简化了将接收器连接器从自管理集群迁移到 Amazon MSK Connect 的过程。

要将接收器连接器迁移到 Amazon MSK Connect，请执行以下操作：

1. 通过从本地或自行管理的 Kafka Connect 集群中提取连接器库，创建 Amazon MSK Connect 自定义插件。
2. 创建 Amazon MSK Connect 工作程序属性，`key.converter`并将`value.converter`属性和值设置为为在现有 Kafka Connect 集群中运行的 Kafka 连接器设置的相同值。
3. 通过在现有 Kafka Connect 集群上`PUT /connectors/`*`connector-name`*`/pause`发出请求，暂停现有集群上的连接器应用程序。
4. 确保连接器应用程序的所有任务都已完全停止。您可以通过在现有 Kafka Connect 集群上`GET /connectors/`*`connector-name`*`/status`发出请求或使用为该属性`status.storage.topic`设置的主题名称中的消息来停止任务。
5. 从现有集群获取连接器配置。您可以通过在现有集群上发出`GET /connectors/`*`connector-name`*`/config`请求或使用为该属性设置的主题名称中的消息来获取连接器配置`config.storage.topic`。
6. 创建一个与现有集群同名的新 Amazon MSK 连接器。使用您在步骤 1 中创建的连接器自定义插件、在步骤 2 中创建的工作器属性和在步骤 5 中提取的连接器配置来创建此连接器。
7. 当 Amazon MSK 连接器状态为时`active`，查看日志以验证连接器是否已开始从源系统导入数据。
8. 通过提出`DELETE /connectors/`*`connector-name`*请求来删除现有集群中的连接器。

排查 Amazon MSK Connect 的问题

以下信息可帮助您排查使用 MSK Connect 时可能存在的问题。您也可以将问题发布到 AWS re:Post。

连接器无法访问公有互联网上托管的资源

请参阅为 Amazon MSK Connect 启用互联网访问。

连接器正在运行的任务数不等于 tasks.max 中指定的任务数量

以下是连接器使用的任务可能少于指定的 tasks.max 配置的一些原因：

- 某些连接器实现限制了可使用的任务数量。例如，适用于 MySQL 的 Debezium 连接器仅限于使用单个任务。
- 使用自动扩缩容量模式时，Amazon MSK Connect 会覆盖连接器的 tasks.max 属性，其值与连接器中运行的工作程序数量和每个工作程序的 MCU 数量成正比。
- 对于接收器连接器，并行度（任务数量）不能超过主题分区的数量。虽然您可以将 tasks.max 设置为大于该值，但单个分区一次只能由一个任务处理。
- 在 Kafka Connect 2.7.x 中，默认的使用器分区分配器是 `RangeAssignor`。该分配器的行为是将每个主题的第一个分区分配给单个使用器，将每个主题的第二个分区分配给单个使用器，依此类推。这意味着，使用 `RangeAssignor` 的接收器连接器的最大活动任务数等于正在消耗的任何单个主题中的最大分区数。如果这不适用于您的用例，则应创建一个工作程序配置，其中将 `consumer.partition.assignment.strategy` 属性设置为更合适的使用器分区分配器。请参阅 Kafka 2.7 接口 ConsumerPartitionAssignor：所有已知实现类。

MSK 复制器

什么是 Amazon MSK 复制器？

Amazon MSK Replicator 是一项 Amazon MSK 功能，它使您能够在不同或相同 AWS 区域的 Amazon MSK 集群之间可靠地复制数据。借助 MSK 复制器，您可以轻松构建具有区域弹性的流媒体应用程序，以提高可用性和业务连续性。MSK 复制器可在 MSK 集群之间提供自动异步复制，无需编写自定义代码、管理基础设施或设置跨区域网络。

MSK 复制器会自动扩缩底层资源，这样您就可以按需复制数据，而无需监控或扩展容量。MSK 复制器还会复制必要的 Kafka 元数据，包括主题配置、访问控制列表（ACL）和使用器组偏移。如果某个区域发生意外事件，您可以故障转移到另一个 AWS 区域并无缝地恢复处理。

MSK 复制器支持跨区域复制（CRR）和同区域复制（SRR）。在跨区域复制中，源和目标 MSK 集群位于不同的 AWS 区域。在同区域复制中，源 MSK 集群和目标 MSK 集群都在同一个 AWS 区域中。在将源和目标 MSK 集群与 MSK 复制器一起使用之前，您需要创建源集群和目标 MSK 集群。

> **Note**
>
> MSK Replicator 支持以下 AWS 区域：美国东部（美国东部 1，弗吉尼亚北部）；美国东部（美国东部 2，俄亥俄州）；美国西部（us-west-2，俄勒冈）；欧洲（eu-west-1，爱尔兰）；欧洲（eu-central-1，法兰克福）；亚太地区（ap-southeast-1，新加坡）；亚太地区（ap-southeast-2，悉尼），欧洲（eu-southeast-1，悉尼）north-1，斯德哥尔摩）、亚太地区（ap-southeast-1，孟买）、欧洲（eu-west-3、巴黎）、南美（sa-east-1，圣保罗）、亚太地区（ap-sontheast-2、伦敦）、亚太地区（ap-northeast-1，东京）、美国西部（us-west-1，加利福尼亚北部）、加拿大（ca-central-1，Central）。

以下是 Amazon MSK 复制器的一些常见用法。

- 构建多区域流媒体应用程序：无需设置自定义解决方案即可构建高度可用且具有容错能力的流媒体应用程序，以提高弹性。
- 更低延迟的数据访问：为不同地理区域的使用器提供更低延迟的数据访问。
- 向合作伙伴分发数据：将数据从一个 Apache Kafka 集群复制到多个 Apache Kafka 集群，这样不同的团队/合作伙伴就可以拥有自己的数据副本。
- 聚合数据进行分析：将来自多个 Apache Kafka 集群的数据复制到一个集群中，以便轻松生成有关聚合实时数据的见解。

- 本地写入，全局访问您的数据：设置多活复制，自动将在一个 AWS 区域执行的写入操作传播到其他区域，从而以更低的延迟和成本提供数据。

Amazon MSK 复制器的工作原理

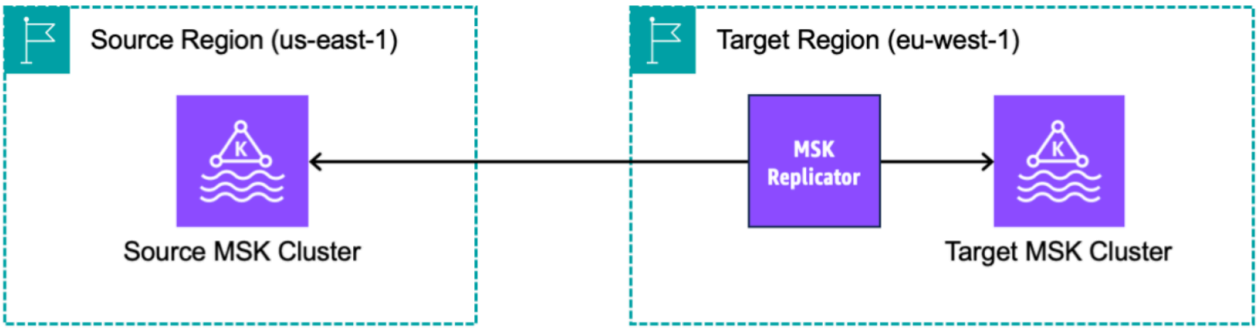
要开始使用 MSK Replicator，你需要在目标集群的区域中创建一个新的复制器。 AWS MSK Replicator 会自动将主 AWS 区域中名为源的集群中的所有数据复制到目标区域中名为目标的集群。源集群和目标集群可以位于相同或不同的 AWS 区域。如果目标集群尚不存在，则需要创建该集群。

当您创建 Replicator 时，MSK Replicator 会在目标集群的 AWS 区域中部署所有必需的资源，以优化数据复制延迟。复制延迟因许多因素而异，包括 MSK 集群 AWS 区域之间的网络距离、源集群和目标集群的吞吐容量以及源集群和目标集群上的分区数量。MSK 复制器会自动扩缩底层资源，这样您就可以按需复制数据，而无需监控或扩展容量。

默认情况下，MSK Replicator 将所有数据和元数据从源集群主题分区的最新偏移异步复制到目标集群。元数据包括主题配置、访问控制列表 (ACL) 和使用者组偏移。如果 “检测并复制新主题” 设置已开启，MSK Replicator 会自动检测新主题并将其复制到目标集群。但是，Replicator 最多可能需要一分钟才能在目标集群上检测和创建新主题。在目标集群上创建主题之前向源主题生成的任何消息都不会被复制。或者，如果您想在目标集群上复制主题上的现有数据和消息，则可以从源集群主题分区中最早的偏移量开始复制。

MSK 复制器不存储您的数据。从源集群中消耗数据，在内存中进行缓冲并写入目标集群。当数据成功写入或重试后失败时，缓冲区会自动清除。MSK Replicator 与您的集群之间的所有通信和数据始终在传输过程中进行加密。

MSK Replicator 在目标集群中创建复制器因子为 3 的主题。如果需要，可以直接在目标集群上修改重复因子。



创建 Amazon MSK 复制器的要求和注意事项

请注意有关运行 Amazon MSK 复制器的这些 MSK 集群要求。

主题

- 授予权限以创建 MSK 复制器
- 支持的集群类型和版本
- MSK Serverless 集群配置
- 集群配置更改

授予权限以创建 MSK 复制器

以下是创建 MSK 复制器所需的 IAM policy 示例。只有在创建 MSK 复制器时提供了标签的情况下，才需要执行 `kafka:TagResource` 操作。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole",
                "iam:CreateServiceLinkedRole",
                "ec2:DescribeSubnets",
```

```
                "ec2:DescribeSecurityGroups",
                "ec2:CreateNetworkInterface",
                "ec2:DescribeVpcs",
                "kafka:CreateReplicator",
                "kafka:TagResource"
            ],
            "Resource": "*"
        }
    ]
}
```

以下是描述复制器的示例 IAM policy。需要 `kafka:DescribeReplicator` 操作或 `kafka:ListTagsForResource` 操作之一即可，而不是两者都需要。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "kafka:DescribeReplicator",
                "kafka:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

支持的集群类型和版本

这些是对支持的实例类型、Kafka 版本和网络配置的要求。

- MSK 复制器支持 MSK 预置集群和 MSK Serverless 集群的任意组合，作为源集群和目标集群。MSK 复制器目前不支持其他类型的 Kafka 集群。
- MSK Serverless 集群需要 IAM 访问控制，不支持 Apache Kafka ACL 复制，并且对主题配置复制的支持有限。请参阅 MSK Serverless。
- 只有运行 Apache Kafka 2.7.0 或更高版本的集群才支持 MSK Replicator，无论您的源集群和目标集群是在相同区域还是位于不同的区域。 AWS
- MSK 复制器支持使用 m5.large 或更大的实例类型的集群。不支持 t3.small 集群。

- 如果您将 MSK 复制器与 MSK 预置集群一起使用，则源集群和目标集群中至少需要三个代理。您可以在两个可用区的集群之间复制数据，但这些集群中至少需要四个代理。
- 您的源 MSK 集群和目标 MSK 集群必须位于同一个 AWS 账户中。不支持跨不同账户的集群复制。
- 如果源集群和目标 MSK 集群位于不同的 AWS 区域（跨区域），则 MSK Replicator 要求源集群为其 IAM 访问控制方法开启多 VPC 私有连接。源集群上的其他身份验证方法不需要使用多 VPC。如果您要在同一 AWS 区域的集群之间复制数据，则不需要多 VPC。请参阅 the section called “单区域中的 多 VPC 私有连接”。
- 如果源集群已开启分层存储，则无法在起始位置设置为 “最早” 的情况下创建 Replicator。

MSK Serverless 集群配置

- MSK Serverless 支持在创建主题期间为 MSK Serverless 目标集群复制以下主题配置：`cleanup.policy`、`compression.type`、`max.message.bytes`、`retention.bytes`、`retenti`
- 在主题配置同步期间，MSK Serverless 仅支持以下主题配置：`compression.type`、`max.message.bytes`、`retention.bytes`、`retention.ms`。
- 复制器在目标 MSK Serverless 集群上使用 83 个压缩分区。确保目标 MSK Serverless 集群有足够数量的压缩分区。请参阅 MSK Serverless 限额。

集群配置更改

- 建议您不要在创建 MSK 复制器后打开或关闭分层存储。如果您的目标集群未分层，则无论您的源集群是否分层，MSK 都不会复制分层存储配置。如果在创建复制器后在目标集群上开启分层存储，则需要重新创建复制器。如果要将数据从非分层集群复制到分层集群，则不应复制主题配置。请参阅在现有主题上启用和禁用分层存储。
- 创建 MSK 复制器后，请勿更改集群配置设置。集群配置设置将在创建 MSK 复制器期间进行验证。为避免 MSK 复制器出现问题，请勿在创建 MSK 复制器后更改以下设置。
  - 将 MSK 集群更改为 t3 实例类型。
  - 更改服务执行角色权限。
  - 禁用 MSK 多 VPC 私有连接。
  - 更改附加的集群基于资源的策略。
  - 更改集群安全组规则。

开始使用 Amazon MSK 复制器

本教程向您展示如何在同一 AWS 区域或不同 AWS 区域中设置源集群和目标集群。然后，您可以使用这些集群创建 Amazon MSK 复制器。

步骤 1：准备 Amazon MSK 源集群

如果您已经为 MSK 复制器创建了 MSK 源集群，请确保它满足本节中描述的要求。否则，请按照以下步骤创建 MSK 预置或无服务器源集群。

创建跨区域和同区域的 MSK 复制器源集群的过程类似。差异将在以下过程中引用。

1. 在源区域中开启了 IAM 访问控制的情况下创建 MSK 预置集群或无服务器集群。您的源集群必须至少有三个代理。
2. 对于跨区域 MSK 复制器，如果源是预置集群，请在为 IAM 访问控制方案开启多 VPC 私有连接的情况下对其进行配置。请注意，开启多 VPC 时，不支持未经身份验证的身份验证类型。您无需为其他身份验证方案（mTLS 或 SASL/SCRAM）开启多 VPC 私有连接。您可以同时对连接到您的 MSK 集群的其他客户端使用 mTLS 或 SASL/SCRAM 身份验证方案。您可以在控制台集群详细信息网络设置中或使用 `UpdateConnectivity` API 配置多 VPC 私有连接。请参阅集群所有者开启多 VPC。如果您的源集群是 MSK Serverless 集群，则无需开启多 VPC 私有连接。

   对于同区域的 MSK 复制器，MSK 源集群不需要多 VPC 私有连接，并且其他客户端仍然可以使用未经身份验证的身份验证类型访问该集群。
3. 对于跨区域 MSK 复制器，您必须将基于资源的权限策略附加到源集群。这允许 MSK 连接到此集群以复制数据。您可以使用下面的 CLI 或 AWS 控制台程序执行此操作。另请参阅 Amazon MSK 基于资源的策略。对于同区域的 MSK 复制器，您不需要执行此步骤。

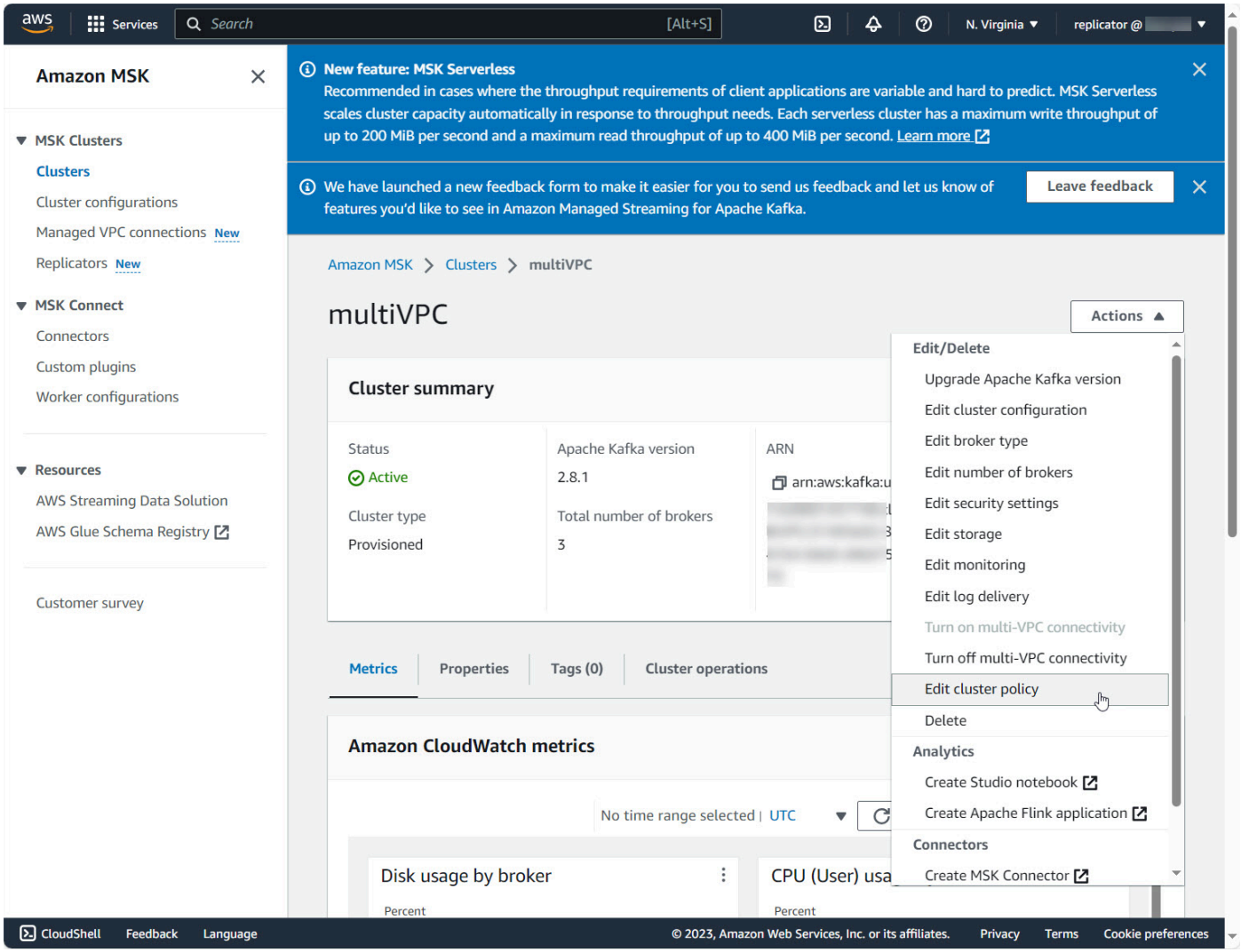
Console: create resource policy

使用以下 JSON 更新源集群策略。将占位符替换为源集群的 ARN。

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Service": [
                "kafka.amazonaws.com"
```

```
            ]
        },
        "Action": [
            "kafka:CreateVpcConnection",
            "kafka:GetBootstrapBrokers",
            "kafka:DescribeClusterV2"
        ],
        "Resource": "<sourceClusterARN>"
    }
  ]
}
```

使用集群详细信息页面上的操作菜单下的编辑集群策略选项。



CLI: create resource policy

注意：如果您使用 AWS 控制台创建源集群并选择创建新 IAM 角色的选项，则会将所需的信任策略 AWS 附加到该角色。另一方面，如果您希望 MSK 使用现有 IAM 角色或您自己创建角色，请将以下信任策略附加到该角色，以便 MSK 复制器可以代入该角色。有关如何修改角色的信任关系的更多信息，请参阅修改角色。

1. 使用此命令获取 MSK 集群策略的当前版本。将占位符替换为实际的集群 ARN。

```
aws kafka get-cluster-policy –cluster-arn <Cluster ARN>
{
"CurrentVersion": "K1PA6795UKM GR7",
"Policy": "..."
}
```

2. 创建基于资源的策略，以允许 MSK 复制器访问您的源集群。使用以下语法作为模板，将占位符替换为实际的源集群 ARN。

```
aws kafka put-cluster-policy --cluster-arn "<sourceClusterARN>" --policy '{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Principal": {
"Service": [
"kafka.amazonaws.com"
]
},
"Action": [
"kafka:CreateVpcConnection",
"kafka:GetBootstrapBrokers",
"kafka:DescribeClusterV2"
],
"Resource": "<sourceClusterARN>"
}
]
```

步骤 2：准备 Amazon MSK 目标集群

在开启了 IAM 访问控制的情况下创建 MSK 目标集群（预置或无服务器集群）。目标集群不需要开启多 VPC 私有连接。目标集群可以与源集群位于同一 AWS 区域或不同的区域。源集群和目标集群必须位于同一个 AWS 账户中。您的目标集群必须至少有三个代理。

步骤 3：创建 Amazon MSK 复制器

在您创建 Amazon MSK 复制器之前，请确保已拥有 授予权限以创建 MSK 复制器。

主题

在目标集群区域使用 AWS 控制台创建复制器

1. 在目标 MSK 集群所在的 AWS 区域，打开 Amazon MSK 控制台，网址为 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。
2. 选择复制器以显示账户中的复制器列表。
3. 选择创建复制器。
4. 在复制器详细信息窗格中，为新的复制器指定一个唯一的名称。

选择源集群

源集群包含要复制到目标 MSK 集群的数据。

1. 在源集群窗格中，选择源集群所在的 AWS 区域。

   您可以通过前往 MSK 集群并查看集群详情 ARN 来查找集群的区域。区域名称嵌入在 ARN 字符串中。在以下示例 ARN 中，`ap-southeast-2` 位于集群区域中。

   ```
   arn:aws:kafka:ap-southeast-2:123456789012:cluster/cluster-11/
   eec93c7f-4e8b-4baf-89fb-95de01ee639c-s1
   ```

2. 输入您的源集群 ARN，或浏览以选择您的源集群。

3. 为您的源集群选择子网。

   控制台显示源集群区域中可用的子网供您选择。必须至少选择两个子网。对于同区域的 MSK 复制器，您选择的用于访问源集群的子网和用于访问目标集群的子网必须位于同一个可用区中。

4. 为 MSK 复制器选择安全组以访问您的源集群。

   - 对于跨区域复制（CRR），请为源集群选择安全组。控制台显示源集群 AWS 区域中可用的安全组供您选择。所选安全组与每个连接相关联。有关使用安全组的更多信息，请参阅 Amazon VPC 用户指南中的使用安全组控制 AWS 资源流量。
   - 对于同区域复制（SRR），请访问 Amazon EC2 控制台（https://console.aws.amazon.com/ec2/）并确保您将为复制器提供的安全组具有入站规则，允许来自源集群安全组的流量。此外，请确保源集群的安全组具有出站规则，允许流量流向为源提供的复制器安全组。

     1. 在 AWS 控制台中，选择集群名称，进入您的 MSK 源集群的详细信息。
     2. 选择属性选项卡，然后向下滚动到网络设置窗格，以选择所应用的安全组名称。
     3. 转到入站规则，然后选择编辑入站规则。
     4. 选择添加规则。
     5. 在新规则中，选择类型列中的所有流量。
     6. 在源列中，键入您将在为源集群创建 复制器期间提供的安全组的名称（这可能与 MSK 源集群的安全组相同），然后选择保存规则。
     7. 选择操作，然后选择编辑出站规则。
     8. 选择添加规则。
     9. 选择类型列中的所有流量。
     10. 在“源”列中，键入 `0.0.0.0/0`，然后选择保存规则。

选择目标集群

目标集群是源数据复制到的 MSK 预置集群或无服务器集群。

> **Note**
>
> MSK 复制器在目标集群中创建新主题，并在主题名称中添加自动生成的前缀。例如，MSK 复制器将“topic”中的数据从源集群复制到目标集群中名为 `<sourceKafkaClusterAlias>.topic` 的新主题。这是为了将包含从源集群复制的数据的主题与目标集群中的其他主题区分开来，并避免在集群之间循环复制数据。您可以使用 DescribeReplicator API 或 MSK 控制台上的 Replicator 详细信息页面在

“sourceKafkaCluster别名” 字段下找到将添加到目标集群中主题名称的前缀。目标集群中的前缀是 < sourceKafkaCluster Alias>。

1. 在目标集群窗格中，选择目标集群所在的 AWS 区域。
2. 输入目标集群的 ARN 或浏览以选择目标集群。
3. 为目标集群选择子网。

   控制台显示目标集群区域中可用的子网供您选择。至少选择两个子网。
4. 为 MSK 复制器选择安全组以访问您的目标集群。

   将显示目标集群区域中可用的安全组供您选择。所选安全组与每个连接相关联。有关使用安全组的更多信息，请参阅 Amazon VPC 用户指南中的使用安全组控制 AWS 资源流量。

   - 对于跨区域复制（CRR），请确保您将在目标集群部分中为 复制器提供的安全组具有出站规则，以允许流量进入目标集群的安全组。此外，请确保目标集群的安全组具有入站规则，以接受来自为目标提供的复制器安全组的流量。
   - 对于同区域复制（SRR），请访问 Amazon EC2 控制台（https://console.aws.amazon.com/ec2/）并确保您将用于创建复制器的安全组具有出站规则，以允许来自目标集群安全组的流量。此外，请确保目标集群的安全组具有入站规则，以接受来自为目标提供的复制器安全组的流量。
     1. 在 AWS 控制台中，转到目标 MSK 集群的网络设置，然后选择所应用的安全组的名称。
     2. 转到入站规则，然后选择编辑入站规则。
     3. 选择添加规则。
     4. 在新规则中，选择类型列中的所有流量。
     5. 在源列中，键入您将在为目标集群创建复制器期间提供的安全组的名称（这可能与目标 MSK 集群的安全组相同），然后选择保存规则。
     6. 转到出站规则，然后选择编辑出站规则。
     7. 选择添加规则。
     8. 选择类型列中的所有流量。
     9. 在源列中，键入 `0.0.0.0/0`，然后选择保存规则。

配置复制器设置和权限

1. 在复制器设置窗格中，使用允许和拒绝列表中的正则表达式指定要复制的主题。默认情况下会复制所有主题。

> **Note**
>
> MSK Replicator 最多只能按排序顺序复制 750 个主题。如果您需要复制更多主题，我们建议您单独创建一个 Replicator。如果您需要为每个 Replicator 提供超过 750 个主题的支持，请前往 AWS 控制台 Support Center 并创建支持案例。您可以使用 “TopicCount” 指标监控正在复制的主题数量。请参阅 Amazon MSK 限额。

2. 默认情况下，MSK Replicator 从选定主题中的最新（最新）偏移量开始复制。或者，如果您要复制主题中的现有数据，则可以从所选主题中最早（最旧）的偏移量开始复制。复制器创建后，您就无法更改此设置。此设置对应于`CreateReplicator`请求和`DescribeReplicator`响应 API 中的`startingPosition`字段。

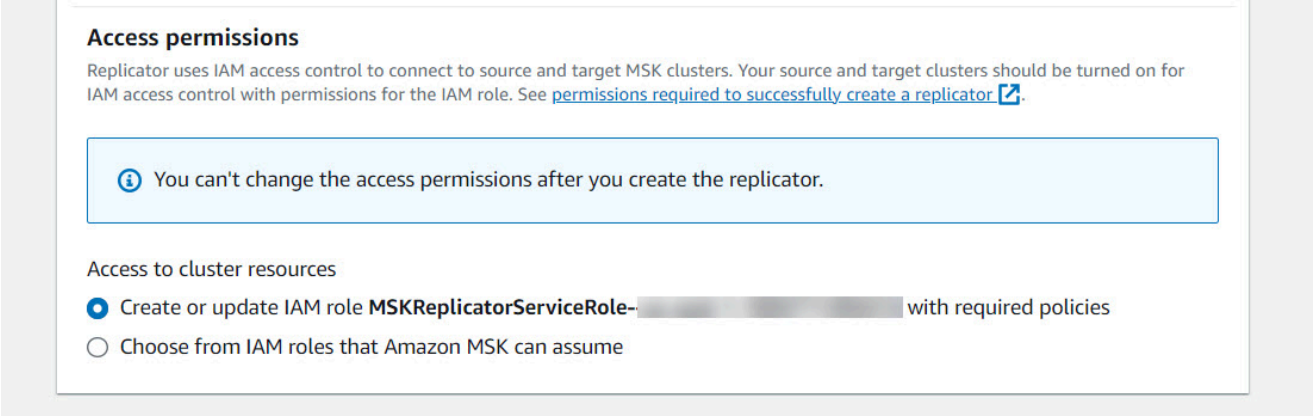
> **Note**
>
> MSK Replicator 就像源集群的新使用者一样。根据您要复制的数据量和源集群中消耗的容量，这可能会导致源集群上的其他使用者受到限制。如果您创建的复制器设置为最早的起始位置，MSK Replicator 将在开始时读取大量数据，这可能会消耗源集群的所有消耗容量。在您的 Replicator 赶上之后，消耗率应降低，以匹配源集群主题的吞吐量。如果您要从最早的位置进行复制，我们建议您使用 Kafka 配额管理 Replicator 吞吐量，以确保其他使用者不会受到限制。

3. 默认情况下，MSK 复制器会复制所有元数据，包括主题配置、访问控制列表（ACL）和使用器组偏移，以实现无缝失效转移。如果您创建的不是用于失效转移的复制器，则可以选择关闭其他设置部分中提供的一个或多个设置。

> **Note**
>
> MSK 复制器不会复制写入 ACL，因为您的生成器不应直接写入目标集群中已复制的主题。失效转移后，您的生成器应写入目标集群中的本地主题。有关详细信息，请参阅 按计划向辅助 AWS 区域执行故障转移。

4. 在使用器组复制窗格中，使用允许和拒绝列表中的正则表达式指定要复制的主题。默认情况下，所有使用器组都会被复制。
5. 在压缩窗格中，您可以选择压缩写入目标集群的数据。如果您要使用压缩，我们建议您使用与源集群中的数据相同的压缩方法。
6. 在访问权限窗格中，执行以下任一操作：

a. 选择使用所需策略创建或更新 IAM 角色。MSK 控制台将自动为服务执行角色附加必要的权限和信任策略，以便读取和写入您的源和目标 MSK 集群。



b. 选择从 Amazon MSK 可以担任的 IAM 角色中选择，提供您自己的 IAM 角色。我们建议您将AWSMSKReplicatorExecutionRole托管 IAM 策略附加到您的服务执行角色，而不是自己编写 IAM 策略。

- 创建 IAM 角色，复制器将使用的该角色对源和目标 MSK 集群进行读取和写入操作，并使用以下 JSON 作为信任策略的一部分，并将 AWSMSKReplicatorExecutionRole 附加到该角色。在信任策略中，将占位符 <yourAccountID> 替换为您的实际账户 ID。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "kafka.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<yourAccountID>"
                }
            }
        }
    ]
}
```

7. 在复制器标签窗格中，您可以选择为 MSK 复制器资源分配标签。有关更多信息，请参阅 为 Amazon MSK 集群添加标签。对于跨区域 MSK 复制器，在创建复制器时，标签会自动同步到远

程区域。如果在创建复制器后更改标签，则更改不会自动同步到远程区域，因此您需要手动同步本地复制器和远程复制器参考。

8. 选择创建。

如果您想限制`kafka-cluster:WriteData`权限，请参阅 A mazon MSK 的 IAM 访问控制的工作原理中的创建授权策略部分。您需要为源集群和目标集群添加`kafka-cluster:WriteDataIdempotently`权限。

大约需要 30 分钟才能成功创建 MSK 复制器并转换到 RUNNING 状态。

如果您创建一个新的 MSK 复制器来替换已删除的复制器，则新的复制器会从最新的偏移开始复制。

如果您的 MSK 复制器已转换为 FAILED 状态，请参阅问题排查部分排查 MSK 复制器的问题。

编辑 MSK 复制器设置

创建 MSK Replicator 后，您就无法更改源集群、目标集群或 Replicator 的起始位置。但是，您可以编辑其他 Replicator 设置，例如要复制的主题和使用者组。

1. 登录并打开亚马逊 MSK 控制台，网址为 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。 AWS Management Console
2. 在左侧导航窗格中，选择复制器以显示账户中的复制器列表，然后选择要编辑的 MSK 复制器。
3. 选择属性选项卡。
4. 在复制器设置部分，选择编辑复制器。
5. 您可以通过更改任一设置来编辑 MSK 复制器设置。

   - 使用允许和拒绝列表中的正则表达式指定要复制的主题。默认情况下，MSK 复制器会复制所有元数据，包括主题配置、访问控制列表（ACL）和使用器组偏移，以实现无缝失效转移。如果您创建的不是用于失效转移的复制器，则可以选择关闭其他设置部分中提供的一个或多个设置。

     > **Note**
     >
     > MSK 复制器不会复制写入 ACL，因为您的生成器不应直接写入目标集群中已复制的主题。失效转移后，您的生成器应写入目标集群中的本地主题。有关详细信息，请参阅 按计划向辅助 AWS 区域执行故障转移。

- 对于使用器组复制，您可以在允许和拒绝列表中使用正则表达式指定要复制的使用器组。默认情况下，所有使用器组都会被复制。如果允许列表和拒绝列表为空，则关闭使用器组复制。
- 在目标压缩类型下，您可以选择压缩写入目标集群的数据。如果您要使用压缩，我们建议您使用与源集群中的数据相同的压缩方法。

6. 保存您的更改。

   大约需要 30 分钟才能成功创建 MSK 复制器并转换到 RUNNING 状态。如果您的 MSK 复制器已转换为 FAILED 状态，请参阅问题排查部分 ???。

删除 MSK 复制器

如果 MSK 复制器创建失败（FAILED 状态），则可能需要将其删除。一旦创建 MSK 复制器，就无法更改分配给 MSK 复制器的源集群和目标集群。您可以删除现有 MSK 复制器并创建新的复制器。如果您创建一个新的 MSK 复制器来替换已删除的复制器，则新的复制器会从最新的偏移开始复制。

1. 在您的源集群所在的 AWS 区域，登录并打开 Amazon MSK 控制台 AWS Management Console，网址为 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。
2. 在导航窗格中，选择复制器。
3. 从 MSK 复制器列表中，选择要删除的复制器，然后选择删除。

监控复制

您可以在目标集群区域中使用 https://console.aws.amazon.com/cloudwatch/ 来查看每个 Amazon MSK 复制器的 `ReplicationLatency`、`MessageLag` 和 `ReplicatorThroughput` 在主题和汇总级别的指标。在 “AWS/Kafka” 命名空间下ReplicatorName方可以看到指标。您还可以查看 `ReplicatorFailure`、`AuthError` 和 `ThrottleTime` 指标来检查问题。

MSK 控制台显示每个 MSK CloudWatch 复制器的指标子集。从控制台复制器列表中，选择复制器的名称并选择监控选项卡。

MSK 复制器指标

以下指标描述了 MSK 复制器的性能或连接指标。

AuthError 指标不包括主题级别的身份验证错误。要监控 MSK Replicator 的主题级身份验证错误，请监控 Replicator 的 ReplicationLatency 指标和源集群的主题级指标。 MessagesInPerSec如果主题

ReplicationLatency 降至 0，但该主题仍有数据正在生成给它，则表示 Replicator 在该主题上存在身份验证问题。检查复制器的服务执行 IAM 角色是否有足够的权限访问该主题。

指标类型	指标	描述	尺寸	单位	原始指标粒度	原始指标聚合统计数据	
Performance	ReplicationLatency	将记录从源集群复制到目标集群所花费的时间；源集群的记录生成时间与复制到目标集群之间的间隔。如果 ReplicationLatency 增加，请检查集群是否有足够的分区来支持复制。当分区数太低而无法实现高吞吐量时，可能会出现较高的复制延迟。	ReplicatorName	毫秒	分区	最高	
			ReplicatorName，话题	毫秒	分区	最高	
Performance	MessageLag	监控 MSK 复制器和源集群之间的同步。MessageLag 表示向源集群生成的消息与复制器使用的消息之间的延迟。这不是源集群和目标集群之间的延迟。即使源集群不可	ReplicatorName	计数	分区	总和	
			ReplicatorName，话题	计数	分区	总和	

指标类型	指标	描述	尺寸	单位	原始指标粒度	原始指标聚合统计数据	
		用/中断，复制器也将完成向目标集群写入已消耗的消息。中断后， MessageLag 显示一个增加的消息，表示复制器位于源集群后面的消息数量，并且可以对其进行监控，直到消息数为 0，这表明复制器已经赶上了源集群。					
Performance	ReplicatorThroughput	每秒复制的平均字节数。如果某个 ReplicatorThroughput 主题被删除，请检查 KafkaClusterPingSuccessCount 和 AuthError 指标以确保 Replicator 可以与集群通信，然后检查集群指标以确保集群没有关闭。	ReplicatorName	BytesPerSecond	分区	总和	
			ReplicatorName，话题	BytesPerSecond	分区	总和	

指标类型	指标	描述	尺寸	单位	原始指标粒度	原始指标聚合统计数据	
Debug	AuthError	每秒身份验证失败的连接数。如果此指标大于 0，则可以检查复制器的服务执行角色策略是否有效，并确保没有为集群权限设置任何拒绝权限。根据 clusterAlias 维度，您可以确定源集群或目标集群是否遇到身份验证错误。	ReplicatorName, ClusterAlias	计数	工作线程	总和	

指标类型	指标	描述	尺寸	单位	原始指标粒度	原始指标聚合统计数据	
Debug	ThrottleTime	集群上的代理限制请求的平均时间（以毫秒为单位）。设置节流以避免 MSK 复制器使集群不堪重负。如果此指标为 0，replicationLatency 不高，并且 replicatorThroughput 符合预期，则表示节流按预期运行。如果该指标大于 0，则可以相应地调整节流。	ReplicatorName, ClusterAlias	毫秒	工作线程	最高	
Debug	ReplicatorFailure	复制器遇到的故障数。	ReplicatorName	计数		总和	

指标类型	指标	描述	尺寸	单位	原始指标粒度	原始指标聚合统计数据	
Debug	KafkaClusterPingSuccessCount	表示与 kafka 集群的复制器连接的运行状况。如果该值为 1，则表示连接正常。如果该值为 0 或没有数据点，则连接不正常。如果该值为 0，则可以检查 Kafka 集群的网络或 IAM 权限设置。根据 ClusterAlias 维度，您可以确定该指标是针对源集群还是目标集群。	ReplicatorName, ClusterAlias	计数		总和	

使用复制来提高 Kafka 流媒体应用程序跨区域的弹性

您可以使用 MSK Replicator 设置主动-主动或主动-被动集群拓扑，以提高 Apache Kafka 应用程序跨区域的弹性。 AWS 在主动-主动设置中，两个 MSK 集群都积极提供读取和写入服务。在主动-被动设置中，一次只有一个 MSK 集群主动提供流媒体数据，而另一个集群处于备用状态。

构建多区域 Apache Kafka 应用程序的注意事项

您的使用器必须能够在不影响下游的情况下重新处理重复的消息。MSK Replicator 会复制可能 at-least-once 导致备用集群中出现重复的数据。当您切换到辅助 AWS 区域时，您的消费者可能会多次处理相同的数据。MSK 复制器会优先处理复制数据而不是使用器偏移，以提高性能。失效转移后，使用器可能会开始从较早的偏移中读取，从而导致重复处理。

生成器和使用器还必须容忍丢失最少的数据。由于 MSK Replicator 异步复制数据，因此当主 AWS 区域开始出现故障时，无法保证所有数据都会复制到辅助区域。您可以使用复制延迟来确定未复制到二级区域的最大数据量。

使用主动-主动与主动-被动集群拓扑

主动-主动集群拓扑提供了几乎为零的恢复时间，并且您的流媒体应用程序能够在多个 AWS 区域同时运行。当一个区域中的集群受损时，连接到另一个区域的集群的应用程序会继续处理数据。

主动-被动设置适用于一次只能在一个 AWS 区域运行的应用程序，或者当您需要更多地控制数据处理顺序时。主动-被动设置比主动-主动设置需要更多的恢复时间，因为您必须在二级区域启动整个主动-被动设置，包括您的生成器和使用器，才能在失效转移后恢复流式传输数据。

创建主动-被动 Kafka 集群设置和复制的主题命名

对于主动-被动设置，我们建议您在两个不同的区域中使用类似的生产者、MSK 集群和消费者（使用相同的消费者组名称）设置。 AWS 两个 MSK 集群必须具有相同的读取和写入容量，以确保可靠的数据复制。您需要创建 MSK 复制器才能将数据从主集群持续复制到备用集群。您还需要将生产者配置为将数据写入同一 AWS 区域中集群的主题中。

为确保您的使用器能够可靠地从备用集群重新启动处理，您需要将使用器配置为使用通配符运算符“.*”从主题中读取数据。例如，MSK Replicator 将 “topic1” 从主集群复制到备用集群中名为 “< Alias>.topic1” 的新主题中。sourceKafkaCluster例如，您可以在两个区域将生成器配置为写入“topic1”，将使用器配置为通过“.*topic1”进行使用。此示例还将包括 footopic1 之类的主题，因此请根据需要调整通配符运算符。

何时故障转移到辅助 AWS 区域

我们建议您使用监控辅助 AWS 区域的复制延迟 CloudWatch。在主 AWS 区域发生服务事件期间，复制延迟可能会突然增加。如果延迟持续增加，请使用 S AWS ervice Health Dashboard 检查主要 AWS 区域中的服务事件。如果发生事件，您可以故障转移到辅助 AWS 区域。

按计划向辅助 AWS 区域执行故障转移

您可以按计划进行故障转移，以测试应用程序在包含源 MSK 集群的主 AWS 区域发生意外事件时的弹性。计划的故障转移不应导致数据丢失。

1. 关闭所有连接到您的源集群的生成器和使用器。

2. 创建新的 MSK 复制器，将数据从二级区域的 MSK 集群复制到主区域中的 MSK 集群。这是将要写入二级区域的数据复制回主区域所必需的，这样您就可以在意外事件结束后对主区域执行失效自动恢复。
3. 在辅助 AWS 区域的目标集群上启动生产者。
4. 请按照以下选项卡之一的步骤操作，具体取决于应用程序的消息排序要求。

No message ordering

如果您的应用程序不需要消息排序，请使用通配符运算符（例如，`topic`）在辅助 AWS 区域中启动从本地（例如）和复制主题（例如`<sourceKafkaClusterAlias>.topic`）读取内容的使用者。 *主题）。

Message ordering

如果您的应用程序需要消息排序，则仅为目标集群上复制的主题（例如`<sourceKafkaClusterAlias>.topic`）启动使用器，而不为本地主题（例如 `topic`）启动使用器。

1. 等待目标 MSK 集群上所有已复制主题的使用器完成所有数据的处理，这样使用器延迟为 0，而处理的记录数也为 0。然后，停止目标集群上已复制主题的使用器。此时，从源 MSK 集群复制到目标 MSK 集群的所有记录都已使用。
2. 在目标 MSK 集群上启动本地主题（例如 `topic`）的使用器。

对辅助 AWS 区域执行计划外故障转移

如果您的源 MSK 群集 AWS 所在的主区域发生服务事件，并且您想暂时将流量重定向到拥有目标 MSK 群集的辅助 AWS 区域，则可以进行计划外故障转移。计划外失效转移可能会导致一些数据丢失。

1. 尝试关闭所有连接到主区域中源 MSK 集群的生成器和使用器。这可能会失败。
2. 启动连接到二级区域中目标 MSK 集群的生成器。
3. 请按照以下选项卡之一的步骤操作，具体取决于应用程序的消息排序要求。

   No message ordering

   如果您的应用程序不需要消息排序，则在目标 AWS 区域启动使用通配符运算符（例如`topic`）同时读取本地（例如`<sourceKafkaClusterAlias>.topic`）和复制主题（例如`.*topic`）的使用者。

Message ordering

1. 仅为目标集群上复制的主题（例如 `<sourceKafkaClusterAlias>.topic`）启动使用器，而不为本地主题（例如 `topic`）启动使用器。
2. 等待目标 MSK 集群上所有已复制主题的使用器完成所有数据的处理，这样偏移延迟为 0，而处理的记录数也为 0。然后，停止目标集群上已复制主题的使用器。此时，从源 MSK 集群复制到目标 MSK 集群的所有记录都已使用。
3. 在目标 MSK 集群上启动本地主题（例如 `topic`）的使用器。

4. 服务事件在主区域结束后，创建一个新的 MSK Replicator，将数据从辅助区域的 MSK 集群复制到主区域中的 MSK 集群，同时将 Replicator 的起始位置设置为最早。这是将要写入二级区域的数据复制回主区域所必需的，这样您就可以在服务事件结束后对主区域执行失效自动恢复。如果您未将 Replicator 的起始位置设置为最早，则在主区域的服务事件期间，您在辅助区域的集群中生成的任何数据都不会复制回主区域的集群。

对主 AWS 区域执行故障恢复

在主 AWS 区域的服务事件结束后，您可以回切到该区域。在失效自动恢复期间将数据复制回主区域时，MSK 复制器会自动跳过以源集群别名为前缀的主题。

如果您执行了计划外的故障切换步骤，则应该已经创建了故障恢复 Replicator，这是从主区域故障转移到辅助区域的最后一步的一部分。

如果您没有执行计划外的故障转移步骤，则在主区域的服务事件结束后，请创建一个新的 MSK Replicator，将数据从辅助区域的 MSK 集群复制到主区域的 MSK 集群，同时将 Replicator 的起始位置设置为最早。这是将要写入二级区域的数据复制回主区域所必需的，这样您就可以在服务事件结束后对主区域执行失效自动恢复。如果您不将 Replicator 的起始位置从其默认值 “最新” 更改为最早，则在主区域的服务事件期间生成到辅助区域集群的任何数据都不会复制回主区域的集群。

只有在从辅助区域的群集复制到主区域的群集已赶上并且中的 MessageLag 指标接近 0 之后，CloudWatch 才应启动故障恢复步骤。计划的失效自动恢复不应导致数据丢失。

1. 关闭所有连接到二级区域中 MSK 集群的生成器和使用器。
2. 对于主动-被动拓扑，请删除正在将数据从二级区域的集群复制到主区域的复制器。对于主动-主动拓扑，您无需删除复制器。
3. 启动连接到主区域中 MSK 集群的生成器。
4. 请按照以下选项卡之一的步骤操作，具体取决于应用程序的消息排序要求。

No message ordering

如果您的应用程序不需要消息排序，则在主 AWS 区域中启动使用通配符运算符（例如`topic`）同时读取本地（例如`<sourceKafkaClusterAlias>.topic`）和复制主题（例如`.*topic`）的使用者。本地主题（例如 topic）的使用器将从失效转移前消耗的最后一个偏移恢复。如果在失效转移之前有任何未处理的数据，则现在将对其进行处理。如果是计划内失效转移，则不应有此类记录。

Message ordering

1. 仅为主区域上复制的主题（例如 `<sourceKafkaClusterAlias>.topic`）启动使用器，而不为本地主题（例如 `topic`）启动使用器。
2. 等待主区域集群上所有已复制主题的使用器完成所有数据的处理，这样偏移延迟为 0，处理的记录数也为 0。然后，停止主区域集群上已复制主题的使用器。此时，失效转移后在二级区域生成的所有记录都已在主区域中使用。
3. 在主区域的集群上启动本地主题（例如 `topic`）的使用器。

5. 使用和延迟指标验证从主区域的群集到辅助区域的群集的现有 Replicator 是否处于 RUNNING 状态`ReplicatorThroughput`并按预期运行。

使用 MSK 复制器创建主动-主动设置

按照以下步骤在源 MSK 集群 A 和目标 MSK 集群 B 之间设置主动-主动拓扑。

1. 创建 MSK 复制器，将 MSK 集群 A 作为源，将 MSK 集群 B 作为目标。
2. 成功创建上述 MSK 复制器后，创建一个以集群 B 为源、集群 A 为目标的复制器。
3. 创建两组生成器，每组生成器将数据同时写入与生成器位于同一区域的集群中的本地主题（例如“主题”）。
4. 创建两组使用者，每组使用通配符订阅读取数据（例如”。 *topic”) 来自与消费者位于同一 AWS 区域的 MSK 集群。这样，您的使用器将自动从本地主题（例如 `topic`）读取在该区域本地生成的数据，以及从主题中带有 `<sourceKafkaClusterAlias>.topic` 前缀的其他区域复制的数据。这两组使用器应具有不同的使用器组 ID，这样当 MSK 复制器将它们复制到另一个集群时，使用器组偏移就不会被覆盖。

排查 MSK 复制器的问题

主题

以下信息可帮助您排查 MSK 复制器可能存在的问题。您也可以将问题发布到 AWS re:Post。

MSK 复制器状态从 CREATING 变为 FAILED

以下是 MSK 复制器创建失败的一些常见原因。

1. 请验证您在目标集群部分中为创建复制器提供的安全组具有出站规则，以允许流量进入目标集群的安全组。此外，请验证目标集群的安全组是否有入站规则，这些规则接受来自您在目标集群部分中为创建复制器提供的安全组的流量。请参阅 选择目标集群。
2. 如果您正在为跨区域复制创建复制器，请确认您的源集群已为 IAM 访问控制身份验证方法开启了多 VPC 连接。请参阅 单区域中的 Amazon MSK 多 VPC 私有连接。此外，请验证是否已在源集群上设置集群策略，以便 MSK 复制器可以连接到源集群。请参阅 步骤 1：准备 Amazon MSK 源集群。
3. 验证您在创建 MSK 复制器期间提供的 IAM 角色是否具有读写源集群和目标集群所需的权限。此外，还要验证 IAM 角色是否具有写入主题的权限。请参阅 配置复制器设置和权限。
4. 确认您的网络 ACL 没有阻止 MSK 复制器与您的源集群和目标集群之间的连接。
5. 当 MSK 复制器尝试连接源集群或目标集群时，源集群或目标集群可能无法完全使用。这可能是由于过度负载、磁盘使用率或 CPU 使用率过高导致复制器无法连接到代理。修复代理的问题，然后重试创建复制器。

执行上述验证后，再次创建 MSK 复制器。

MSK 复制器似乎停留在 CREATING 状态

有时，MSK 复制器创建可能需要长达 30 分钟。请等待 30 分钟，然后再次检查集群的状态。

MSK 复制器没有复制数据或只复制部分数据

请按照以下步骤排查数据复制问题。

1. 使用 MSK Replicator 在中提供的 AuthError 指标，验证您的 Replicator 没有遇到任何身份验证错误。 CloudWatch如果此指标大于 0，请检查您为复制器提供的 IAM 角色的策略是否有效，并且没

有为集群权限设置拒绝权限。根据 clusterAlias 维度，您可以确定源集群或目标集群是否遇到身份验证错误。

2. 请验证您的源集群和目标集群没有遇到任何问题。复制器可能无法连接到您的源集群或目标集群。这可能是由于连接过多、磁盘容量已满或 CPU 使用率过高所致。
3. 使用中的 KafkaClusterPingSuccessCount 指标验证您的源集群和目标集群是否可以从 MSK Replicator 访问。 CloudWatch根据 clusterAlias 维度，您可以确定源集群或目标集群是否遇到身份验证错误。如果该指标为 0 或没有数据点，则连接不正常。您应该检查 MSK 复制器用于连接到集群的网络和 IAM 角色权限。
4. 使用中的指标，验证您的 Replicator 是否没有因为缺少主题级权限而出现故障。 ReplicatorFailure CloudWatch如果此指标大于 0，请检查您为主题级别权限提供的 IAM 角色。
5. 验证您在创建复制器时在允许列表中提供的正则表达式是否与您要复制的主题的名称相匹配。此外，请确认主题没有因为拒绝列表中的正则表达式而被排除在复制之外。

复制延迟很高或持续增加

以下是复制延迟较高的一些常见原因。

1. 验证源和目标 MSK 集群上的分区数量是否正确。分区过少或过多会影响性能。有关选择分区数量的指导，请参阅 使用 MSK 复制器的最佳实践。下表显示要使用 MSK 复制器实现所需吞吐量的建议最小分区数。

   吞吐量和建议的最小分区数

吞吐量（MB/s）	需要的最小分区数量
50	167
100	334
250	833
500	1666
1000	3333

2. 验证您的源和目标 MSK 集群中是否有足够的读取和写入容量来支持复制流量。MSK 复制器充当源集群（出口）的使用器，也充当目标集群（入口）的生成器。因此，除了集群上的其他流量外，您还应预置集群容量以支持复制流量。有关调整 MSK 集群大小的指导，请参阅 ???。

3. 不同源和目标 AWS 区域对中的 MSK 集群的复制延迟可能会有所不同，具体取决于集群在地理上相隔的距离。例如，与欧洲地区（爱尔兰）和亚太地区（悉尼）区域的集群之间的复制相比，在欧洲地区（爱尔兰）和欧洲地区（伦敦）区域的集群之间进行复制时，复制延迟通常较低。
4. 验证复制器没有因为在源集群或目标集群上设置的限额过于激进而受到限制。您可以使用 MSK Replicator 提供的 ThrottleTime指标 CloudWatch 来查看源/目标集群上代理限制请求的平均时间（以毫秒为单位）。如果此指标大于 0，则应调整 Kafka 限额以减少节流，以便复制器能够赶上。有关管理复制器的 Kafka 限额的信息，请参阅 使用 Kafka 限额管理 MSK 复制器吞吐量。
5. ReplicationLatency 当一个 AWS 区域退化时， MessageLag 可能会增加。使用 AWS 服务运行状况控制面板查看您的主 MSK 集群所在的区域中是否有 MSK 服务事件。如果有服务事件，可以临时将应用程序的读取和写入重定向到另一个区域。

使用 MSK 复制器的最佳实践

本节介绍使用 MSK 复制器的常见最佳实践和实现策略。

主题

- 使用 Kafka 限额管理 MSK 复制器吞吐量
- 设置集群保留期

使用 Kafka 限额管理 MSK 复制器吞吐量

由于 MSK 复制器充当源集群的使用器，复制可能会导致源集群上的其他使用器受到节流。节流量取决于源集群的读取容量和要复制的数据吞吐量。我们建议您为源集群和目标集群配置相同的容量，并在计算所需容量时考虑复制吞吐量。

您还可以在源集群和目标集群上为复制器设置 Kafka 限额，以控制 MSK 复制器可以使用的容量。建议使用网络带宽限额。网络带宽限额定义了一个或多个共享限额的客户端的字节速率阈值，定义为每秒字节数。此限额依不同代理而定义。

请按照以下步骤申请限额。

1. 检索源集群的引导服务器字符串。请参阅 获取 Amazon MSK 集群的引导代理。
2. 检索 MSK 复制器使用的服务执行角色（SER）。这是您用于 `CreateReplicator` 请求的 SER。您也可以从现有复制器的 DescribeReplicator 响应中提取 SER。
3. 使用 Kafka CLI 工具，对源集群运行以下命令。

```
./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. 执行上述命令后，请确认该 ReplicatorThroughput 指标未超过您设置的限额。

请注意，如果您在多个 MSK 复制器之间重复使用服务执行角色，则它们都受此限额的约束。如果要为每个复制器保留单独的限额，请使用不同的服务执行角色。

有关使用带限额的 MSK IAM 身份验证的更多信息，请参阅 Multi-tenancy Apache Kafka clusters in Amazon MSK with IAM access control and Kafka Quotas – Part 1。

> **Warning**
>
> 设置极低的 consumer_byte_rate 可能会导致 MSK 复制器以意外的方式运行。

设置集群保留期

您可以为 MSK 预置的集群和无服务器集群设置日志保留期。默认的保留期为 7 天。请参阅 集群配置更改 或 MSK Serverless 集群配置。

集群状态

下表显示了集群的可能状态并描述了这些状态的含义。它还描述了当集群处于其中一种状态时，您可以执行和不能执行的操作。要了解集群的状态，可以访问 AWS Management Console。也可以使用 describe-cluster-v2 命令或 DescribeClusterV2 操作来描述集群。集群的描述包括其状态。

集群状态	含义和可行的操作
ACTIVE (处于活动状态)	您可以生成和使用数据。您还可以在集群上执行 Amazon MSK API 和 AWS CLI 操作。
CREATING	Amazon MSK 正在设置集群。您必须等待集群处于 ACTIVE 状态，才能使用它来生成或使用数据，或者对其执行 Amazon MSK API 或 AWS CLI 操作。
DELETING	正在删除集群。您不能用它来生成或使用数据。您也不能对其执行 Amazon MSK API 和 AWS CLI 操作。
FAILED	集群创建或删除过程失败。您不能用集群来生成或使用数据。您可以删除集群，但无法对其执行 Amazon MSK API 或 AWS CLI 更新操作。
HEALING	Amazon MSK 正在运行内部操作，例如更换运行不正常的代理。例如，代理可能没有响应。您仍可以用集群来生成或使用数据。但是，在集群恢复到 ACTIVE 状态之前，您无法对其执行 Amazon MSK API 或 AWS CLI 更新操作。
MAINTENANCE	Amazon MSK 正在对集群执行例行维护操作。此类维护操作包括安全修补。您仍可以用集群来生成或使用数据。但是，在集群恢复到 ACTIVE 状态之前，您无法对其执行 Amazon MSK API 或 AWS CLI 更新操作。
REBOOTING_BROKER	Amazon MSK 正在重启代理。您仍可以用集群来生成或使用数据。但是，在集群恢复到

集群状态	含义和可行的操作
	ACTIVE 状态之前，您无法对其执行 Amazon MSK API 或 AWS CLI 更新操作。
UPDATING	用户启动的 Amazon MSK API 或 AWS CLI 操作正在更新集群。您仍可以用集群来生成或使用数据。但是，在集群恢复到 ACTIVE 状态之前，您无法对其执行任何其他 Amazon MSK API 或 AWS CLI 更新操作。

Amazon Managed Streaming for Apache Kafka 中的安全性

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 AWS 的责任。责任共担模式将其描述为云的安全性和云中的安全性：

- 云安全 — AWS 负责保护在 AWS 云中运行 AWS 服务的基础架构。 AWS 还为您提供可以安全使用的服务。作为AWS 合规计划合规计划合规计划合的一部分，第三方审计师定期测试和验证我们安全的有效性。要了解适用于 Amazon Managed Streaming for Apache Kafka 的合规性计划，请参阅按合规性计划提供的范围内 Amazon Web Services。
- 云端安全-您的责任由您使用的 AWS 服务决定。您还需要对其它因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

该文档可帮助您了解如何在使用 Amazon MSK 时应用责任共担模式。以下主题说明如何配置 Amazon MSK 以实现您的安全性与合规性目标。您还会了解如何使用其他 Amazon Web Services 帮助您监控和保护 Amazon MSK 资源。

主题

- Amazon Managed Streaming for Apache Kafka 中的数据保护
- Amazon MSK API 的身份验证和授权
- Apache Kafka API 的身份验证和授权
- 更改 Amazon MSK 集群的安全组
- 控制对 Apache 的访问权限 ZooKeeper
- 日志记录
- Amazon Managed Streaming for Apache Kafka 的合规性验证
- Amazon Managed Streaming for Apache Kafka 中的恢复能力
- Amazon Managed Streaming for Apache Kafka 中的基础设施安全性

Amazon Managed Streaming for Apache Kafka 中的数据保护

分担责任模式 AWS 分担责任模型适用于适用于 Apache Kafka 的 Apache Streaming for Apache Streaming 中的数据保护。如本模型所述 AWS ，负责保护运行所有内容的全球基础架构 AWS

Cloud。您负责维护对托管在此基础设施上的内容的控制。您还负责您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题。有关欧洲数据保护的信息，请参阅 AWS 安全性博客 上的 AWS 责任共担模式和 GDPR 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户 凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。我们还建议您通过以下方式保护数据：

- 对每个账户使用 multi-factor authentication（MFA）。
- 使用 SSL/TLS 与资源通信。 AWS 我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-2 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准（FIPS）第 140-2 版》。

我们强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括您使用控制台、API 或软件开发工具包 AWS 服务 使用 Amazon M AWS SK 或其他软件开发工具包的情况。 AWS CLI在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供网址，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

主题

- Amazon MSK 加密
- 如何开始使用加密？

Amazon MSK 加密

Amazon MSK 提供了数据加密选项，可使用这些选项来满足严格的数据管理要求。Amazon MSK 用于加密的证书必须每 13 个月续订一次。Amazon MSK 会自动为所有集群续订这些证书。在它启动证书更新操作时，它会将集群状态设置为 `MAINTENANCE`。待更新完成后，它会将集群状态重新设置为 `ACTIVE`。当集群处于 `MAINTENANCE` 状态时，您可以继续生成和使用数据，但无法对数据执行任何更新操作。

静态加密

Amazon MSK 与 AWS Key Management Service（KMS）集成以提供透明的服务器端加密。Amazon MSK 始终加密您的静态数据。当创建 MSK 集群时，您可以指定您希望 Amazon MSK 用于加密静态数据的 AWS KMS key 。如果您不指定 KMS 密钥，Amazon MSK 会为您创建一个 AWS 托管式密钥并代表您使用它。有关 KMS 密钥的更多信息，请参阅《AWS Key Management Service 开发人员指南》中的 AWS KMS keys。

传输中加密

Amazon MSK 会使用 TLS 1.2。默认情况下，它会加密在 MSK 集群的代理之间传输的数据。可以在创建集群时覆盖此默认值。

对于客户端和代理之间的通信，您必须指定下列三项设置之一：

- 仅允许 TLS 加密数据。这是默认设置。
- 同时允许明文数据和 TLS 加密数据。
- 仅允许明文数据。

亚马逊 MSK 经纪人使用公共 AWS Certificate Manager 证书。因此，任何信任 Amazon Trust Services 的信任库也会信任 Amazon MSK 代理的证书。

虽然我们强烈建议启用传输中加密，但它可能会增加额外的 CPU 开销和几毫秒的延迟。但是，大多数使用案例对这些差异并不敏感，影响的程度取决于集群、客户端和使用情况配置文件的配置。

如何开始使用加密？

创建 MSK 集群时，您可以使用 JSON 格式指定加密设置。以下是示例。

```
{
   "EncryptionAtRest": {
       "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/abcdabcd-1234-
abcd-1234-abcd123e8e8e"
    },
   "EncryptionInTransit": {
        "InCluster": true,
        "ClientBroker": "TLS"
    }
}
```

对于 DataVolumeKMSKeyId，您可以为账户 (alias/aws/kafka) 中的 MSK 指定客户托管密钥或 AWS 托管式密钥 。如果您未指定EncryptionAtRest，Amazon MSK 仍会对您的静态数据进行加密。 AWS 托管式密钥要确定您的集群使用的密钥，请发送 GET 请求或调用 DescribeCluster API 操作。

对于 EncryptionInTransit，InCluster 的默认值为 true，但是如果您不想在代理之间传递数据时让 Amazon MSK 加密数据，则可以将此项设置为 false。

要为客户端和代理之间传输的数据指定加密模式，请将 ClientBroker 设置为以下三个值之一：TLS、TLS_PLAINTEXT 或 PLAINTEXT。

创建集群时指定加密设置

1. 将上一示例的内容保存在文件中，并为该文件指定所需的任何名称。例如，将其命名为 encryption-settings.json。
2. 运行 create-cluster 命令并使用 encryption-info 选项指向您保存配置 JSON 的文件。以下是示例。将 *{YOUR MSK VERSION}* 替换为与 Apache Kafka 客户端版本相匹配的版本。有关如何查找 MSK 集群版本的信息，请参阅 To find the version of your MSK cluster。请注意，使用与 MSK 集群版本不同的 Apache Kafka 客户端版本，可能会导致 Apache Kafka 数据损坏、丢失和停机。

```
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
```

以下是运行此命令后的成功响应示例。

```
{
    "ClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/SecondTLSTest/
abcdabcd-1234-abcd-1234-abcd123e8e8e",
    "ClusterName": "ExampleClusterName",
    "State": "CREATING"
}
```

测试 TLS 加密

1. 按照the section called “步骤 3：创建客户端计算机”中的指导创建客户端计算机。
2. 在客户端计算机上安装 Apache Kafka。

3. 在 AWS CLI 安装了的计算机上运行以下命令，将 *clusterArn* 替换为您的集群（与前面过程中的示例一样，使用TLS设置为ClientBroker创建的集群）的 ARN。

```
aws kafka describe-cluster --cluster-arn clusterARN
```

在结果中，查找 ZookeeperConnectString 的值并保存它，因为您需要在下一步中使用该值。

4. 在您的客户端计算机上运行以下命令以创建主题。*ZookeeperConnectString*替换为您在上一步ZookeeperConnectString中获得的值。

```
<path-to-your-kafka-installation>/bin/kafka-topics.sh --create --
zookeeper ZookeeperConnectString --replication-factor 3 --partitions 1 --topic
 TLSTestTopic
```

5. 在本示例中，我们使用 JVM 信任库与 MSK 集群通信。为此，请首先在客户端计算机上创建一个名为 /tmp 的文件夹。然后，转到 Apache Kafka 安装的 bin 文件夹，并运行以下命令。（您的 JVM 路径可能不相同。）

```
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
```

6. 仍在客户端计算机上的 Apache Kafka 安装的 bin 文件夹中，创建一个名为 client.properties 的文本文件，该文件包含以下内容。

```
security.protocol=SSL
ssl.truststore.location=/tmp/kafka.client.truststore.jks
```

7. 在 AWS CLI 安装了的计算机上运行以下命令，将 c *lusterArn* 替换为集群的 ARN。

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

成功结果如下所示。保存此结果，因为您需要在下一步中使用它。

```
{
    "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"
}
```

8. 运行以下命令，在客户端计算机上创建控制台生成器。*BootstrapBrokerStringTls*用您在上一步中获得的值替换。保持运行此生成器命令。

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

9. 打开新的命令窗口并连接到同一台客户端计算机。然后，运行以下命令以创建控制台使用器。

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBrokerStringTls --consumer.config client.properties --topic
 TLSTestTopic
```

10. 在生成器窗口中，输入文本消息后点击回车键，并在使用器窗口中查找相同消息。Amazon MSK 对传输中的此消息进行了加密。

有关配置 Apache Kafka 客户端以使用加密数据的更多信息，请参阅配置 Kafka 客户端。

Amazon MSK API 的身份验证和授权

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以通过身份验证（登录）和获得授权（具有权限）来使用 Amazon MSK 资源。您可以使用 IAM AWS 服务 ，无需支付额外费用。

本页介绍如何使用 IAM 来控制谁可以在您的集群上执行 Amazon MSK 操作。有关如何控制谁可以在您的集群上执行 Apache Kafka 操作的信息，请参阅 the section called “Apache Kafka API 的身份验证和授权”。

主题

- Amazon MSK 如何与 IAM 配合使用
- Amazon MSK 基于身份的策略示例
- 对 Amazon MSK 使用服务相关角色
- AWS 亚马逊 MSK 的托管策略
- Amazon MSK 身份和访问问题排查

Amazon MSK 如何与 IAM 配合使用

在使用 IAM 管理对 Amazon MSK 的访问权限之前，您应该了解哪些 IAM 功能可用于 Amazon MSK。要全面了解 Amazon MSK 和其他 AWS 服务如何与 IAM 配合使用，请参阅 IAM 用户指南中的与 IAM 配合使用的AWS 服务。

主题

- Amazon MSK 基于身份的策略
- Amazon MSK 基于资源的策略
- AWS 托管策略
- 基于 Amazon MSK 标签的授权
- Amazon MSK IAM 角色

Amazon MSK 基于身份的策略

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。Amazon MSK 支持特定的操作、资源和条件键。要了解在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》 中的 IAM JSON 策略元素参考。

操作

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体 可以对什么资源 执行操作，以及在什么 条件 下执行。

JSON 策略的 `Action` 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 AWS API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限 操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行关联操作的权限。

Amazon MSK 中的策略操作在操作前使用以下前缀：`kafka:`。例如，要授予某人使用 Amazon MSK `DescribeCluster` API 操作描述 MSK 集群的权限，您应将 `kafka:DescribeCluster` 操作纳入其策略。策略语句必须包含 `Action` 或 `NotAction` 元素。Amazon MSK 定义了自己的一组操作，以描述您可以使用该服务执行的任务。

要在单个语句中指定多项操作，请使用逗号将它们隔开，如下所示：

```
"Action": ["kafka:action1", "kafka:action2"]
```

您也可以使用通配符（*）指定多个操作。例如，要指定以单词 Describe 开头的所有操作，包括以下操作：

```
"Action": "kafka:Describe*"
```

要查看 Amazon MSK 操作的列表，请参阅《IAM 用户指南》中的 Amazon Managed Streaming for Apache Kafka 的操作、资源和条件键。

资源

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体 可以对什么资源 执行操作，以及在什么条件 下执行。

Resource JSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 Resource 或 NotResource 元素。作为最佳实践，请使用其 Amazon 资源名称（ARN）指定资源。对于支持特定资源类型（称为资源级权限）的操作，您可以执行此操作。

对于不支持资源级权限的操作（如列出操作），请使用通配符 (*) 指示语句应用于所有资源。

```
"Resource": "*"
```

Amazon MSK 实例资源具有以下 ARN：

```
arn:${Partition}:kafka:${Region}:${Account}:cluster/${ClusterName}/${UUID}
```

有关 ARN 格式的更多信息，请参阅 A mazon 资源名称 (ARN) 和 AWS 服务命名空间。

例如，要在语句中指定 CustomerMessages 实例，请使用以下 ARN：

```
"Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/CustomerMessages/abcd1234-
abcd-dcba-4321-a1b2abcd9f9f-2"
```

要指定属于特定账户的所有实例，请使用通配符 (*)：

```
"Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/*"
```

无法对特定资源执行某些 Amazon MSK 操作，例如用于创建资源的操作。在这些情况下，您必须使用通配符（*）。

```
"Resource": "*"
```

要在单个语句中指定多个资源，请使用逗号分隔 ARN。

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

要查看 Amazon MSK 资源类型及其 ARN 的列表，请参阅《IAM 用户指南》中的 Resources Defined by Amazon Managed Streaming for Apache Kafka。要了解您可以在哪些操作中指定每个资源的 ARN，请参阅 Amazon Managed Streaming for Apache Kafka 定义的操作。

条件键

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

在 Condition 元素（或 Condition 块）中，可以指定语句生效的条件。Condition 元素是可选的。您可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。

如果您在一个语句中指定多个 Condition 元素，或在单个 Condition 元素中指定多个键，则 AWS 使用逻辑 AND 运算评估它们。如果您为单个条件键指定多个值，则使用逻辑OR运算来 AWS 评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，您也可以使用占位符变量。例如，只有在使用 IAM 用户名标记 IAM 用户时，您才能为其授予访问资源的权限。有关更多信息，请参阅 IAM 用户指南中的 IAM policy 元素：变量和标签。

AWS 支持全局条件密钥和特定于服务的条件密钥。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的AWS 全局条件上下文密钥。

Amazon MSK 定义了自己的一组条件键，还支持使用一些全局条件键。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的AWS 全局条件上下文密钥。

要查看 Amazon MSK 条件键的列表，请参阅《IAM 用户指南》中的 Amazon Managed Streaming for Apache Kafka 的条件键。要了解您可以对哪些操作和资源使用条件键，请参阅 Amazon Managed Streaming for Apache Kafka 定义的操作。

示例

要查看 Amazon MSK 基于身份的策略的示例，请参阅 Amazon MSK 基于身份的策略示例。

Amazon MSK 基于资源的策略

Amazon MSK 支持用于 Amazon MSK 集群的集群策略（也称为基于资源的策略）。您可以使用集群策略来定义哪些 IAM 主体拥有跨账户权限来设置与 Amazon MSK 集群的私有连接。当与 IAM 客户端身份验证一起使用时，您也可以使用集群策略为连接的客户端精细定义 Kafka 数据面板的权限。

要查看如何配置集群策略的示例，请参阅 步骤 2：将集群策略附加到 MSK 集群。

AWS 托管策略

基于 Amazon MSK 标签的授权

您可以将标签附加到 Amazon MSK 集群。要基于标签控制访问，您需要使用 `kafka:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的条件元素中提供标签信息。有关标记 Amazon MSK 资源的更多信息，请参阅 the section called “为集群添加标签”。

要查看基于身份的策略（用于基于集群上的标签来限制对该集群的访问）的示例，请参阅根据标签访问 Amazon MSK 集群。

Amazon MSK IAM 角色

IAM 角色是 Amazon Web Services 账户中具有特定权限的实体。

将临时凭证用于 Amazon MSK

可以使用临时凭证进行联合身份验证登录，分派 IAM 角色或分派跨账户角色。您可以通过调用AssumeRole或之类的 AWS STS API 操作来获取临时安全证书GetFederationToken。

Amazon MSK 支持使用临时凭证。

服务相关角色

服务相关角色允许 Amazon Web Services 访问其他服务中的资源，以代表您完成操作。服务相关角色显示在 IAM 账户中，并归该服务所有。管理员可以查看但不能编辑服务相关角色的权限。

Amazon MSK 支持服务相关角色。有关创建或管理 Amazon MSK 服务相关角色的详细信息，请参阅 the section called “服务相关角色”。

Amazon MSK 基于身份的策略示例

默认情况下，IAM 用户和角色无权执行 Amazon MSK API 操作。管理员必须创建 IAM policy，以便为用户和角色授予权限以对所需的指定资源执行特定的 API 操作。然后，管理员必须将这些策略附加到需要这些权限的 IAM 用户或组。

要了解如何使用这些示例 JSON 策略文档创建 IAM 基于身份的策略，请参阅《IAM 用户指南》中的 在 JSON 选项卡上创建策略。

主题

策略最佳实践

基于身份的策略确定某个人是否可以创建、访问或删除您账户中的 Amazon MSK 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下准则和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的 AWS 托管式策略 或 工作职能的AWS 托管式策略。
- 应用最低权限 – 在使用 IAM policy 设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。
- 使用 IAM policy 中的条件进一步限制访问权限 – 您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定的方式使用的，则也可以使用条件来授予对服务操作的访问权限 AWS 服务，例如 AWS CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件。
- 使用 IAM Access Analyzer 验证您的 IAM policy，以确保权限的安全性和功能性 – IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM policy 语言 (JSON) 和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的 IAM Acess Analyzer 策略验证。

- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的 配置受 MFA 保护的 API 访问。

有关 IAM 中的最佳实操的更多信息，请参阅《IAM 用户指南》中的 IAM 中的安全最佳实操。

允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管式策略。此策略包括在控制台上或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

```
{
    "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": "*"
        }
    ]
```

```
}
```

访问一个 Amazon MSK 集群

在此示例中，您想要为 Amazon Web Services 账户中的 IAM 用户授予访问某个集群 `purchaseQueriesCluster` 的权限。此策略允许用户描述集群、获取其引导代理、列出其代理节点并更新它。

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

根据标签访问 Amazon MSK 集群

您可以在基于身份的策略中使用条件，以便基于标签控制对 Amazon MSK 资源的访问权限。此示例演示了如何创建允许用户描述集群、获取其引导代理、列出其代理节点、更新和删除集群的策略。但是，仅当集群标签 `Owner` 的值为该用户的用户名时，才能授予此权限。

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

您可以将该策略附加到您账户中的 IAM 用户。如果名为 `richard-roe` 的用户尝试更新 MSK 集群，必须将集群标记为 `Owner=richard-roe` 或 `owner=richard-roe`。否则，他将被拒绝访问。条件标签键 `Owner` 匹配 `Owner` 和 `owner`，因为条件键名称不区分大小写。有关更多信息，请参阅 IAM 用户指南 中的 IAM JSON 策略元素：条件。

对 Amazon MSK 使用服务相关角色

Amazon MSK 使用 AWS Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种独特类型的 IAM 角色，它与 Amazon MSK 直接关联。服务相关角色由 Amazon MSK 预定义，包括该服务代表您调用其他 AWS 服务所需的所有权限。

服务相关角色可让您更轻松地设置 Amazon MSK，因为您不必手动添加所需权限。Amazon MSK 可定义其服务相关角色的权限。除非另行定义，否则只有 Amazon MSK 才能代入其角色。定义的权限包括信任策略和权限策略，而且权限策略不能附加到任何其它 IAM 实体。

有关支持服务相关角色的其他服务的信息，请参阅使用 IAM 的 Amazon Web Services，并查找服务相关角色列中显示为是的服务。选择是，可转到查看该服务的服务相关角色文档的链接。

主题

- Amazon MSK 的服务相关角色权限
- 为 Amazon MSK 创建服务相关角色
- 为 Amazon MSK 编辑服务相关角色
- Amazon MSK 服务相关角色支持的区域

Amazon MSK 的服务相关角色权限

Amazon MSK 使用名为 AWSServiceRoleForKafka 的服务相关角色。Amazon MSK 使用此角色来访问您的资源并执行以下操作：

- `*NetworkInterface` – 在客户账户中创建和管理网络接口，使客户 VPC 中的客户端可以访问集群代理。
- `*VpcEndpoints`— 管理客户账户中的 VPC 终端节点，这些终端节点允许客户 VPC 中的客户使用集群代理 AWS PrivateLink。Amazon MSK 对 `DescribeVpcEndpoints`、`ModifyVpcEndpoint` 和 `DeleteVpcEndpoints` 使用权限。
- `secretsmanager`— 使用管理客户凭证 AWS Secrets Manager。
- `GetCertificateAuthorityCertificate` – 检索私有证书颁发机构的证书。

此服务相关角色附加到以下托管策略：`KafkaServiceRolePolicy`。有关本政策的更新，请参阅KafkaServiceRolePolicy。

AWSServiceRoleForKafka 服务相关角色信任以下服务代入该角色：

- `kafka.amazonaws.com`

角色权限策略允许 Amazon MSK 对资源完成以下操作。

```
{
 "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_*"
    }
   }
  }
 ]
}
```

您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的服务相关角色权限。

为 Amazon MSK 创建服务相关角色

您无需手动创建服务相关角色。当您在 AWS Management Console、或 AWS API 中创建 Amazon MSK 集群时 AWS CLI，Amazon MSK 会为您创建服务相关角色。

如果您删除该服务相关角色，然后需要再次创建，您可以使用相同流程在账户中重新创建此角色。当您创建 Amazon MSK 集群时，Amazon MSK 将再次为您创建服务相关角色。

为 Amazon MSK 编辑服务相关角色

Amazon MSK 不允许您编辑 AWSServiceRoleForKafka 服务相关角色。创建服务相关角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。但是可以使用 IAM 编辑角色描述。有关更多信息，请参阅《IAM 用户指南》中的编辑服务相关角色。

Amazon MSK 服务相关角色支持的区域

Amazon MSK 支持在该服务可用的所有区域中使用服务相关角色。有关更多信息，请参阅 AWS 区域和端点。

AWS 亚马逊 MSK 的托管策略

AWS 托管策略是由创建和管理的独立策略 AWS。 AWS 托管策略旨在为许多常见用例提供权限，以便您可以开始为用户、组和角色分配权限。

请记住， AWS 托管策略可能不会为您的特定用例授予最低权限权限，因为它们可供所有 AWS 客户使用。我们建议通过定义特定于您的使用场景的客户管理型策略来进一步减少权限。

您无法更改 AWS 托管策略中定义的权限。如果 AWS 更新 AWS 托管策略中定义的权限，则更新会影响该策略所关联的所有委托人身份（用户、组和角色）。 AWS 最有可能在启动新的 API 或现有服务可以使用新 AWS 服务 的 API 操作时更新 AWS 托管策略。

有关更多信息，请参阅《IAM 用户指南》中的 AWS 托管式策略。

AWS 托管策略：AmazonMSK FullAccess

此策略授予管理权限，允许主体完全访问所有 Amazon MSK 操作。此策略中的权限分组如下：

- Amazon MSK 权限允许所有 Amazon MSK 操作。
- **Amazon EC2**权限 — 在此策略中，需要在 API 请求中验证传递的资源。这是为了确保 Amazon MSK 能够成功在集群中使用资源。此策略中的其他 Amazon EC2 权限允许 Amazon MSK 创建必要的 AWS 资源，使您能够连接到您的集群。

- **AWS KMS**权限 — 在 API 调用期间用于验证请求中传递的资源。Amazon MSK 必须使用它们才能在 Amazon MSK 集群中使用传递的密钥。
- **CloudWatch Logs, Amazon S3, and Amazon Data Firehose**权限 — Amazon MSK 需要权限，才能确保日志传输目标可访问，并且这些目标对代理日志的使用有效。
- **IAM**权限 — Amazon MSK 需要权限，才能在您的账户中创建服务相关角色并允许您将服务执行角色传递给 Amazon MSK。

```
{
 "Version": "2012-10-17",
 "Statement": [{
   "Effect": "Allow",
   "Action": [
    "kafka:*",
    "ec2:DescribeSubnets",
    "ec2:DescribeVpcs",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeRouteTables",
    "ec2:DescribeVpcEndpoints",
    "ec2:DescribeVpcAttribute",
    "kms:DescribeKey",
    "kms:CreateGrant",
    "logs:CreateLogDelivery",
    "logs:GetLogDelivery",
    "logs:UpdateLogDelivery",
    "logs:DeleteLogDelivery",
    "logs:ListLogDeliveries",
    "logs:PutResourcePolicy",
    "logs:DescribeResourcePolicies",
    "logs:DescribeLogGroups",
    "S3:GetBucketPolicy",
    "firehose:TagDeliveryStream"
   ],
   "Resource": "*"
  },
  {
   "Effect": "Allow",
   "Action": [
    "ec2:CreateVpcEndpoint"
   ],
   "Resource": [
    "arn:*:ec2:*:*:vpc/*",
```

```
        "arn:*:ec2:*:*:subnet/*",
        "arn:*:ec2:*:*:security-group/*"
       ]
      },
      {
       "Effect": "Allow",
       "Action": [
        "ec2:CreateVpcEndpoint"
       ],
       "Resource": [
        "arn:*:ec2:*:*:vpc-endpoint/*"
       ],
       "Condition": {
        "StringEquals": {
         "aws:RequestTag/AWSMSKManaged": "true"
        },
        "StringLike": {
         "aws:RequestTag/ClusterArn": "*"
        }
       }
      },
      {
       "Effect": "Allow",
       "Action": [
        "ec2:CreateTags"
       ],
       "Resource": "arn:*:ec2:*:*:vpc-endpoint/*",
       "Condition": {
        "StringEquals": {
         "ec2:CreateAction": "CreateVpcEndpoint"
        }
       }
      },
      {
       "Effect": "Allow",
       "Action": [
        "ec2:DeleteVpcEndpoints"
       ],
       "Resource": "arn:*:ec2:*:*:vpc-endpoint/*",
       "Condition": {
        "StringEquals": {
         "ec2:ResourceTag/AWSMSKManaged": "true"
        },
        "StringLike": {
```

```
          "ec2:ResourceTag/ClusterArn": "*"
         }
        }
       },
       {
        "Effect": "Allow",
        "Action": "iam:PassRole",
        "Resource": "*",
        "Condition": {
         "StringEquals": {
          "iam:PassedToService": "kafka.amazonaws.com"
         }
        }
       },
       {
        "Effect": "Allow",
        "Action": "iam:CreateServiceLinkedRole",
        "Resource": "arn:aws:iam::*:role/aws-service-role/kafka.amazonaws.com/
AWSServiceRoleForKafka*",
        "Condition": {
         "StringLike": {
          "iam:AWSServiceName": "kafka.amazonaws.com"
         }
        }
       },
       {
        "Effect": "Allow",
        "Action": [
         "iam:AttachRolePolicy",
         "iam:PutRolePolicy"
        ],
        "Resource": "arn:aws:iam::*:role/aws-service-role/kafka.amazonaws.com/
AWSServiceRoleForKafka*"
       },
       {
        "Effect": "Allow",
        "Action": "iam:CreateServiceLinkedRole",
        "Resource": "arn:aws:iam::*:role/aws-service-role/delivery.logs.amazonaws.com/
AWSServiceRoleForLogDelivery*",
        "Condition": {
         "StringLike": {
          "iam:AWSServiceName": "delivery.logs.amazonaws.com"
         }
        }
```

```
      }

     ]
    }
```

AWS 托管策略：AmazonMSK ReadOnlyAccess

此策略授予只读权限，允许用户查看 Amazon MSK 中的信息。附加此策略的主体不能进行任何更新或删除现有资源，也不能创建新的 Amazon MSK 资源。例如，拥有这些权限的主体可以查看与其账户关联的集群和配置列表，但不能更改任何集群的配置或设置。此策略中的权限分组如下：

- **Amazon MSK**权限 — 允许您列出 Amazon MSK 资源，对其进行描述并获取有关它们的信息。
- **Amazon EC2**权限 — 用于描述与集群关联的 Amazon VPC、子网、安全组和 ENI。
- **AWS KMS**权限 — 用于描述与集群关联的密钥。

```
{
    "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 托管策略： KafkaServiceRolePolicy

您无法附加 KafkaServiceRolePolicy 到您的 IAM 实体。将此策略附加到服务相关角色，该角色允许 Amazon MSK 执行诸如管理 MSK 集群上的 VPC 端点（连接器）、管理网络接口和使用 AWS Secrets Manager管理集群凭证等操作。有关更多信息，请参阅 the section called “服务相关角色”。

AWS 托管策略： AWSMSKReplicatorExecutionRole

该AWSMSKReplicatorExecutionRole策略向 Amazon MSK 复制器授予在 MSK 集群之间复制数据的权限。此策略中的权限如下分组：

- **cluster**— 向 Amazon MSK Replicator 授予使用 IAM 身份验证连接到集群的权限。还授予描述和更改集群的权限。
- **topic**— 授予 Amazon MSK Replicator 描述、创建和更改主题以及更改主题动态配置的权限。
- **consumer group**— 授予 Amazon MSK Replicator 描述和更改使用者组、从 MSK 集群读取和写入日期以及删除复制器创建的内部主题的权限。

```
{
 "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/*/*"
   ]
  }
 ]
}
```

亚马逊 MSK 更新了托 AWS 管政策

查看自该服务开始跟踪这些更改以来，Amazon MSK AWS 托管政策更新的详细信息。

更改	描述	日期
WriteDataIdempotently 权限已添加到 AWSMSKReplicatorExecutionRole-更新现有策略	Amazon MSK 为 AWSMSKReplicatorExecutionRole 策略添加了支持 MSK 集群之间数据复制的 WriteDataIdempotently 权限。	2024 年 3 月 12 日
AWSMSKReplicatorExecutionRole：新策略	亚马逊 MSK 增加了支持 Amazon MSK Replicator	2023 年 12 月 4 日

更改	描述	日期
	的 AWSMSKReplicatorExecutionRole 政策。	
AmazonMSK FullAccess — 更新现有政策	Amazon MSK 添加了支持 Amazon MSK 复制器的权限。	2023 年 9 月 28 日
KafkaServiceRolePolicy – 更新了现有策略	Amazon MSK 添加了支持多 VPC 私有连接的权限。	2023 年 3 月 8 日
AmazonMSK FullAccess — 更新现有政策	Amazon MSK 添加了新的 Amazon EC2 权限，以便连接到集群。	2021 年 11 月 30 日
AmazonMSK FullAccess — 更新现有政策	Amazon MSK 添加了新权限，允许其描述 Amazon EC2 路由表。	2021 年 11 月 19 日
Amazon MSK 开启了跟踪更改	Amazon MSK 开始跟踪其 AWS 托管政策的变更。	2021 年 11 月 19 日

Amazon MSK 身份和访问问题排查

使用以下信息可帮助您诊断和修复在使用 Amazon MSK 和 IAM 时可能遇到的常见问题。

主题

- 我无权在 Amazon MSK 中执行操作

我无权在 Amazon MSK 中执行操作

如果 AWS Management Console 告诉您您无权执行某项操作，则必须联系管理员寻求帮助。管理员是向您提供登录凭证的人。

当 `mateojackson` IAM 用户尝试使用控制台删除集群，但没有 `kafka:DeleteCluster` 权限时，会发生以下示例错误。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 kafka:DeleteCluster on resource: purchaseQueriesCluster
```

在这种情况下，Mateo 请求他的管理员更新其策略，以允许他使用 `kafka:DeleteCluster` 操作访问 `purchaseQueriesCluster` 资源。

Apache Kafka API 的身份验证和授权

您可以使用 IAM 对客户端进行身份验证并允许或拒绝 Apache Kafka 操作。或者，您也可以使用 TLS 或 SASL/SCRAM 对客户端进行身份验证，以及使用 Apache Kafka ACL 来允许或拒绝操作。

有关如何控制谁可以在您的集群上执行 Amazon MSK 操作的信息，请参阅 the section called “Amazon MSK API 的身份验证和授权”。

主题

- IAM 访问控制
- 双向 TLS 身份验证
- 使用 S AWS ecrets Manager 进行登录凭据身份验证
- Apache Kafka ACL

IAM 访问控制

Amazon MSK 的 IAM 访问控制让您能够处理 MSK 集群的身份验证和授权。这样就不需要使用一种身份验证机制和另一种授权机制。例如，当客户端尝试写入您的集群时，Amazon MSK 使用 IAM 来检查该客户端是否是经过身份验证的身份，以及是否有权向您的集群生成数据。IAM 访问控制适用于 Java 和非 Java 客户端，包括用 Python、 JavaScript Go 和.NET 编写的 Kafka 客户端。

Amazon MSK 会记录访问事件，以方便您进行审计。有关更多信息，请参阅 the section called “CloudTrail 事件”。

为了能够进行 IAM 访问控制，Amazon MSK 对 Apache Kafka 源代码进行了少许修改。这些修改不会给您的 Apache Kafka 体验造成明显的影响。

> **Important**
>
> IAM 访问控制不适用于 Apache ZooKeeper 节点。有关如何控制对这些节点的访问权限的信息，请参阅 the section called “控制对 Apache 的访问权限 ZooKeeper”。

> **Important**
>
> 如果您的集群使用 IAM 访问控制，则 `allow.everyone.if.no.acl.found` Apache Kafka 设置无效。

> **Important**
>
> 您可以为使用 IAM 访问控制的 MSK 集群调用 Apache Kafka ACL API。但是，存储在 Apache 中的 Apache Kafka ACL 对 IAM 角色的授权 ZooKeeper 没有影响。您必须将 IAM 策略用于 IAM 角色的访问控制。

Amazon MSK 的 IAM 访问控制的工作原理

要使用 Amazon MSK 的 IAM 访问控制，请执行以下步骤，本节的其余部分将详细介绍这些步骤。

- the section called “创建使用 IAM 访问控制的集群”
- the section called “配置客户端以进行 IAM 访问控制”
- the section called “创建授权策略”
- the section called “获取用于 IAM 访问控制的引导代理”

创建使用 IAM 访问控制的集群

本节介绍如何使用 AWS Management Console、API 或创建使用 IAM 访问控制的集群。 AWS CLI 有关如何为现有集群开启 IAM 访问控制的信息，请参阅 the section called “更新安全设置”。

使用创建 AWS Management Console 使用 IAM 访问控制的集群

1. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
2. 选择创建集群。
3. 选择使用自定义设置创建集群。
4. 在身份验证部分中，选择 IAM 访问控制。
5. 完成创建集群的其余工作流程。

使用 API 或创建 AWS CLI 使用 IAM 访问控制的集群

- 要创建启用了 IAM 访问控制的集群，请使用 CreateClusterAPI 或 create-cluster CLI 命令，并传递以下 JSON 作为ClientAuthentication参数：。"ClientAuthentication": { "Sasl": { "Iam": { "Enabled": true } }

配置客户端以进行 IAM 访问控制

要让客户端能够与使用 IAM 访问控制的 MSK 集群通信，您可以使用以下任何一种机制：

- 使用 SASL_OAUTHBEARER 机制进行非 Java 客户端配置
- 使用 SASL_OAUTHBEARER 机制或 AWS_MSK_IAM 机制进行 Java 客户端配置

使用 SASL_OAUTHBEARER 机制配置 IAM

1. 使用以下 Python Kafka 客户端示例中突出显示的语法作为指南，编辑 client.properties 配置文件。其他语言的配置更改与之类似。

```
#!/usr/bin/python3from kafka import KafkaProducer
from kafka.errors import KafkaError
import socket
import time
from aws_msk_iam_sasl_signer import MSKAuthTokenProvider

class MSKTokenProvider():
    def token(self):
        token, _ = MSKAuthTokenProvider.generate_auth_token('<my aws region>')
        return token

tp = MSKTokenProvider()

producer = KafkaProducer(
    bootstrap_servers='<my bootstrap string>',
    security_protocol='SASL_SSL',
    sasl_mechanism='OAUTHBEARER',
    sasl_oauth_token_provider=tp,
    client_id=socket.gethostname(),
)

topic = "<my-topic>"
while True:
```

```
    try:
        inp=input(">")
        producer.send(topic, inp.encode())
        producer.flush()
        print("Produced!")
    except Exception:
        print("Failed to send message:", e)

producer.close()
```

2. 下载所选配置语言的帮助程序库，然后按照该语言库主页上的 “开始使用” 部分中的说明进行操作。
   - JavaScript: https://github.com/aws/ aws-msk-iam-sasl-signer-js #getting-st arted
   - Python：https://github.com/aws/ aws-msk-iam-sasl-signer-python #get- 已开始
   - Go：https://github.com/aws/ aws-msk-iam-sasl-signer-go #getting-st arted
   - .NET：https://github.com/aws/ aws-msk-iam-sasl-signer- net #getting-已开始
   - JAVA：Java 通过 `aws-msk-iam-auth` jar 文件提供 SASL_OAUTHBEARER 支持

使用 MSK 自定义 AWS_MSK_IAM 机制配置 IAM

1. 将以下内容添加到 `client.properties` 文件中。将 *<PATH_TO_TRUST_STORE_FILE>* 替换为客户端上信任存储文件的完全限定路径。

   > **Note**
   >
   > 如果您不想使用特定证书，可以从 `client.properties` 文件中删除 `ssl.truststore.location=`*<PATH_TO_TRUST_STORE_FILE>*。如果您不指定 `ssl.truststore.location` 的值，Java 进程将使用默认证书。

   ```
   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
   ```

   要使用您为 AWS 凭证创建的命名配置文件，请将其包含awsProfileName="*your profile name*";在您的客户端配置文件中。有关命名配置文件的信息，请参阅文档中的命名配置 AWS CLI 文件。

2. 下载最新的稳定版 aws-msk-iam-authJAR 文件，并将其放在类路径中。如果您使用 Maven，请添加以下依赖项，并根据需要调整版本号：

```
<dependency>
    <groupId>software.amazon.msk</groupId>
    <artifactId>aws-msk-iam-auth</artifactId>
    <version>1.0.0</version>
</dependency>
```

Amazon MSK 客户端插件在 Apache 2.0 许可证下是开源的。

创建授权策略

将授权策略附加到与客户端对应的 IAM 角色。在授权策略中，您可以指定角色允许或拒绝哪些操作。如果您的客户端位于 Amazon EC2 实例上，请将授权策略与该 Amazon EC2 实例的 IAM 角色关联。或者，您也可以将客户端配置为使用命名配置文件，然后将授权策略与该命名配置文件的角色关联。the section called “配置客户端以进行 IAM 访问控制” 介绍如何将客户端配置为使用命名配置文件。

有关如何创建 IAM policy 的信息，请参阅创建 IAM policy。

以下是名为的集群的授权策略示例 MyTestCluster。要了解 `Action` 和 `Resource` 元素的语义，请参阅 the section called “操作和资源的语义”。

> **Important**
>
> 您对 IAM policy 所做的更改会立即反映在 IAM API 和 AWS CLI 中。但是，策略更改可能需要很长时间才能生效。在大多数情况下，策略更改会在不到一分钟的时间内生效。网络状况有时可能会增加延迟时间。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:AlterCluster",
```

```
                "kafka-cluster:DescribeCluster"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:0123456789012:cluster/MyTestCluster/
abcd1234-0123-abcd-5678-1234abcd-1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:*Topic*",
                "kafka-cluster:WriteData",
                "kafka-cluster:ReadData"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:0123456789012:topic/MyTestCluster/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeGroup"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:0123456789012:group/MyTestCluster/*"
            ]
        }
    ]
}
```

要了解如何创建包含与常见 Apache Kafka 用例（例如创建和使用数据）对应的操作元素的策略，请参阅 the section called “常见使用案例”。

对于 Kafka 版本 2.8.0 及更高版本，该WriteDataIdempotently权限已被弃用 (KIP-679)。默认情况下，设置了 `enable.idempotence = true`。因此，对于 Kafka 版本 2.8.0 及更高版本，IAM 不提供与 Kafka ACL 相同的功能。仅提供 `WriteData` 对某个主题的访问权限，无法 `WriteDataIdempotently` 到该主题。这不会影响将 `WriteData` 提供给所有主题的情况。在这种情况下，允许 `WriteDataIdempotently`。这是由于 IAM 逻辑的实现与 Kafka ACL 的实现方式存在差异。

要解决这个问题，建议使用类似于以下示例的策略：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:AlterCluster",
                "kafka-cluster:DescribeCluster",
                "kafka-cluster:WriteDataIdempotently"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:0123456789012:cluster/MyTestCluster/
abcd1234-0123-abcd-5678-1234abcd-1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:*Topic*",
                "kafka-cluster:WriteData",
                "kafka-cluster:ReadData"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:0123456789012:topic/MyTestCluster/
abcd1234-0123-abcd-5678-1234abcd-1/TestTopic"
            ]
        }
    ]
}
```

在这种情况下，`WriteData` 允许写入 `TestTopic`，而 `WriteDataIdempotently` 允许对集群进行幂等性写入。请务必注意，`WriteDataIdempotently` 是集群级别的权限。它不能在主题级别使用。如果 `WriteDataIdempotently` 仅限于主题级别，则此策略将不起作用。

获取用于 IAM 访问控制的引导代理

请参阅 the section called “获取引导代理”。

操作和资源的语义

本部分解释了可以在 IAM 授权策略中使用的操作和资源元素的语义。有关策略示例，请参阅the section called “创建授权策略”。

操作

下表列出了在使用 Amazon MSK 的 IAM 访问控制时可以在授权策略中包含的操作。当您在授权策略中包含表操作列中的操作时，还必须包含所需操作列中的相应操作。

操作	描述	所需的操作	所需的 资源	适用于无服务器集群
`kafka-cluster:Connect`	授予连接和验证集群的权限。	无	cluster	支持
`kafka-cluster:DescribeCluster`	授予描述集群各个方面的权限，相当于 Apache Kafka 的 DESCRIBE CLUSTER ACL。	`kafka-cluster:Connect`	cluster	支持
`kafka-cluster:AlterCluster`	授予更改集群各个方面的权限，相当于 Apache Kafka 的 ALTER CLUSTER ACL。	`kafka-cluster:Connect` `kafka-cluster:DescribeCluster`	cluster	不支持
`kafka-cluster:DescribeClusterDynamicConfiguration`	授予描述集群动态配置的权限，相当于 Apache Kafka 的 DESCRIBE_CONFIGS CLUSTER ACL。	`kafka-cluster:Connect`	cluster	不支持

操作	描述	所需的操作	所需的 资源	适用于无服务器集群
kafka-cluster:AlterClusterDynamicConfiguration	授予更改集群动态配置的权限，相当于 Apache Kafka 的 ALTER_CONFIGS CLUSTER ACL。	kafka-cluster:Connect kafka-cluster:DescribeClusterDynamicConfiguration	cluster	不支持
kafka-cluster:WriteDataIdempotently	授予在集群上以幂等方式写入数据的权限，相当于 Apache Kafka 的 IDEMPOTENT_WRITE CLUSTER ACL。	kafka-cluster:Connect kafka-cluster:WriteData	cluster	支持
kafka-cluster:CreateTopic	授予在集群上创建主题的权限，相当于 Apache Kafka 的 CREATE CLUSTER/TOPIC ACL。	kafka-cluster:Connect	topic	支持
kafka-cluster:DescribeTopic	授予描述集群上主题的权限，相当于 Apache Kafka 的 DESCRIBE TOPIC ACL。	kafka-cluster:Connect	topic	支持

操作	描述	所需的操作	所需的 资源	适用于无服务器集群
kafka-cluster:AlterTopic	授予更改集群上主题的权限，相当于 Apache Kafka 的 ALTER TOPIC ACL。	kafka-cluster:Connect kafka-cluster:DescribeTopic	topic	支持
kafka-cluster:DeleteTopic	授予删除集群上主题的权限，相当于 Apache Kafka 的 DELETE TOPIC ACL。	kafka-cluster:Connect kafka-cluster:DescribeTopic	topic	支持
kafka-cluster:DescribeTopicDynamicConfiguration	授予描述集群上主题动态配置的权限，相当于 Apache Kafka 的 DESCRIBE_CONFIGS TOPIC ACL。	kafka-cluster:Connect	topic	支持
kafka-cluster:AlterTopicDynamicConfiguration	授予更改集群上主题动态配置的权限，相当于 Apache Kafka 的 ALTER_CONFIGS TOPIC ACL。	kafka-cluster:Connect kafka-cluster:DescribeTopicDynamicConfiguration	topic	支持

操作	描述	所需的操作	所需的 资源	适用于无服务器集群
kafka-cluster:ReadData	授予从集群上主题中读取数据的权限，相当于 Apache Kafka 的 READ TOPIC ACL。	kafka-cluster:Connect kafka-cluster:DescribeTopic kafka-cluster:AlterGroup	topic	支持
kafka-cluster:WriteData	授予向集群上的主题写入数据的权限，相当于 Apache Kafka 的 WRITE TOPIC ACL	kafka-cluster:Connect kafka-cluster:DescribeTopic	topic	支持
kafka-cluster:DescribeGroup	授予描述集群上群组的权限，相当于 Apache Kafka 的 DESCRIBE GROUP ACL。	kafka-cluster:Connect	组	支持
kafka-cluster:AlterGroup	授予加入集群上群组的权限，相当于 Apache Kafka 的 READ GROUP ACL。	kafka-cluster:Connect kafka-cluster:DescribeGroup	组	支持

操作	描述	所需的操作	所需的 资源	适用于无服务器集群
kafka-cluster:DeleteGroup	授予删除集群上群组的权限，相当于 Apache Kafka 的 DELETE GROUP ACL。	kafka-cluster:Connect kafka-cluster:DescribeGroup	组	支持
kafka-cluster:DescribeTransactionalId	授予描述集群上事务 ID 的权限，相当于 Apache Kafka 的 DESCRIBE TRANSACTIONAL_ID ACL。	kafka-cluster:Connect	transactional-id	支持
kafka-cluster:AlterTransactionalId	授予更改集群上事务 ID 的权限，相当于 Apache Kafka 的 WRITE TRANSACTIONAL_ID ACL。	kafka-cluster:Connect kafka-cluster:DescribeTransactionalId kafka-cluster:WriteData	transactional-id	支持

在冒号之后的操作中，您可以任意次数地使用星号 (*) 通配符。示例如下。

- kafka-cluster:*Topic 代表 kafka-cluster:CreateTopic、kafka-cluster:DescribeTopic、kafka-cluster:AlterTopic 和 kafka-cluster:DeleteTopic。它不包括 kafka-

`cluster:DescribeTopicDynamicConfiguration` 或 `kafka-cluster:AlterTopicDynamicConfiguration`。

- `kafka-cluster:*` 代表所有权限。

资源

下表显示了在使用 Amazon MSK 的 IAM 访问控制时可在授权策略中使用的四种资源。您可以使用 DescribeClusterAPI 或 d es AWS CLI cribe-cluster 命令从 AWS Management Console 或中获取集群 Amazon 资源名称 (ARN)。然后，您可以使用集群 ARN 来构造主题、组和事务 ID ARN。要在授权策略中指定资源，请使用该资源的 ARN。

资源	ARN 格式
集群	arn:aws:kafka:*region*:*account-id* :cluster/*cluster-name* /*cluster-uuid*
主题	arn:aws:kafka:*region*:*account-id* :topic/*cluster-name* /*cluster-uuid* /*topic-name*
组	arn:aws:kafka:*region*:*account-id* :group/*cluster-name* /*cluster-uuid* /*group-name*
事务 ID	arn:aws:kafka:*region*:*account-id* :transactional-id/*cluster-name* /*cluster-uuid* /*transactional-id*

您可以在 ARN 中 `:cluster/`、`:topic/`、`:group/` 和 `:transactional-id/` 之后的任意位置，任意次数地使用星号 (*) 通配符。以下是如何使用星号 (*) 通配符引用多个资源的部分示例：

- `arn:aws:kafka:us-east-1:0123456789012:topic/MyTestCluster/*`：任何名为的集群中的所有主题 MyTestCluster，无论集群的 UUID 如何。
- `arn:aws:kafka:us-east-1:0123456789012:topic/MyTestCluster/abcd1234-0123-abcd-5678-1234abcd-1/*_test`：集群中名称以 “_test” 结尾的所有主题，其名称为，UUID 为 abcd1234-0123-abcd-56 MyTestCluster 78-1234abcd-1。
- `arn:aws:kafka:us-east-1:0123456789012:transactional-id/MyTestCluster/*/5555abcd-1111-abcd-1234-abcd1234-1`：所有交易 ID 为 5555abcd-1111-abcd-1234-abcd-1234-1 的交易，涉及你账户中命名的集群的所有化身。 MyTestCluster 这意味着，如果您创建了一个名为 MyTestCluster的集群，然后将其删除，然后创建另一个同名集群，则可以使用此资源 ARN 在两个集群上表示相同的交易 ID。但是，无法访问已删除的集群。

常见使用案例

下表中的第一列显示了一些常见用例。要授权客户端执行给定用例，请在客户端的授权策略中包含该用例所需的操作，并将 `Effect` 设置为 `Allow`。

有关 Amazon MSK 的 IAM 访问控制之所有操作的信息，请参阅 the section called “操作和资源的语义”。

> **Note**
>
> 默认情况下，操作将被拒绝。您必须明确允许要授权客户端执行的每个操作。

应用场景	所需的操作
Admin	`kafka-cluster:*`
创建主题	`kafka-cluster:Connect` `kafka-cluster:CreateTopic`

生成数据	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:WriteData`

使用数据	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:DescribeGroup`
	`kafka-cluster:AlterGroup`
	`kafka-cluster:ReadData`

以幂等方式生成数据	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:WriteData`

应用场景	所需的操作
	kafka-cluster:WriteDataIdempotently
以事务方式生成数据	kafka-cluster:Connect kafka-cluster:DescribeTopic kafka-cluster:WriteData kafka-cluster:DescribeTransactionalId kafka-cluster:AlterTransactionalId
描述集群的配置	kafka-cluster:Connect kafka-cluster:DescribeClusterDynamicConfiguration
更新集群的配置	kafka-cluster:Connect kafka-cluster:DescribeClusterDynamicConfiguration kafka-cluster:AlterClusterDynamicConfiguration
描述主题的配置	kafka-cluster:Connect kafka-cluster:DescribeTopicDynamicConfiguration

应用场景	所需的操作
更新主题的配置	kafka-cluster:Connect kafka-cluster:DescribeTopicDynamicConfiguration kafka-cluster:AlterTopicDynamicConfiguration
更改主题	kafka-cluster:Connect kafka-cluster:DescribeTopic kafka-cluster:AlterTopic

双向 TLS 身份验证

对于从应用程序到 Amazon MSK 代理和 ZooKeeper 节点的连接，您可以使用 TLS 启用客户端身份验证。要使用客户端身份验证，您需要有 AWS 私有 CA。 AWS 私有 CA 可以与您的集群位于 AWS 账户 同一个账户中，也可以位于不同的账户中。有关 AWS 私有 CA s 的信息，请参阅创建和管理 AWS 私有 CA。

> **Note**
>
> TLS 身份验证目前在北京和宁夏区域不可用。

Amazon MSK 不支持证书吊销列表（CRL）。要控制对集群主题的访问权限或屏蔽已泄露的证书，请使用 Apache Kafka ACL 和 AWS 安全组。有关使用 Apache Kafka ACL 的信息，请参阅 the section called “Apache Kafka ACL”。

本主题包含下列部分：

- 创建支持客户端身份验证的集群
- 将客户端设置为使用身份验证
- 使用身份验证生成和使用消息

创建支持客户端身份验证的集群

此过程向您展示如何使用启用客户端身份验证 AWS 私有 CA。

> **Note**
>
> 在使用双向 TLS 控制访问时，我们强烈建议 AWS 私有 CA 对每个 MSK 集群使用独立模式。这样做可以确保由 PCA 签署的 TLS 证书仅在单个 MSK 集群中进行身份验证。

1. 使用以下内容创建名为 `clientauthinfo.json` 的文件。将 *Private-CA-ARN* 替换为您的 PCA 的 ARN。

```
{
   "Tls": {
       "CertificateAuthorityArnList": ["Private-CA-ARN"]
    }
}
```

2. 创建一个名为 `brokernodegroupinfo.json` 的文件，如the section called “使用创建集群 AWS CLI”中所述。
3. 客户端身份验证还要求您启用客户端和代理之间的传输中加密。使用以下内容创建名为 `encryptioninfo.json` 的文件。将 *KMS-Key-ARN* 替换为您的 KMS 密钥的 ARN。可以将 `ClientBroker` 设置为 `TLS` 或 `TLS_PLAINTEXT`。

```
{
   "EncryptionAtRest": {
       "DataVolumeKMSKeyId": "KMS-Key-ARN"
    },
   "EncryptionInTransit": {
        "InCluster": true,
        "ClientBroker": "TLS"
    }
}
```

   有关加密的更多信息，请参阅the section called “加密”。
4. 在 AWS CLI 安装了身份验证和传输中加密的计算机上，运行以下命令来创建集群。保存响应中提供的集群 ARN。

```
aws kafka create-cluster --cluster-name "AuthenticationTest" --broker-node-group-
info file://brokernodegroupinfo.json --encryption-info file://encryptioninfo.json
 --client-authentication file://clientauthinfo.json --kafka-version "{YOUR KAFKA
 VERSION}" --number-of-broker-nodes 3
```

将客户端设置为使用身份验证

1. 创建用作客户端计算机的 Amazon EC2 实例。为简单起见，请在用于集群的同一 VPC 中创建此实例。有关如何创建此类客户端计算机的示例，请参阅the section called “步骤 3：创建客户端计算机”。
2. 创建主题。有关示例，请参阅the section called “步骤 4：创建主题”下的说明。
3. 在已 AWS CLI 安装的计算机上，运行以下命令以获取集群的引导代理。将 *Cluster-ARN* 替换为您的集群的 ARN。

   ```
   aws kafka get-bootstrap-brokers --cluster-arn Cluster-ARN
   ```

   保存与响应中的 BootstrapBrokerStringTls 关联的字符串。
4. 在客户端计算机上，运行以下命令以使用 JVM 信任存储来创建客户端信任存储。如果您的 JVM 路径不同，请相应地调整命令。

   ```
   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. 在客户端计算机上，运行以下命令为客户端创建私有密钥。将 *Distinguished-Name*、*Example-Alias*、*Your-Store-Pass* 和 *Your-Key-Pass* 替换为所选字符串。

   ```
   keytool -genkey -keystore kafka.client.keystore.jks -validity 300 -storepass Your-
   Store-Pass -keypass Your-Key-Pass -dname "CN=Distinguished-Name" -alias Example-
   Alias -storetype pkcs12
   ```
6. 在客户端计算机上，运行以下命令以使用您在上一步中创建的私有密钥创建证书请求。

   ```
   keytool -keystore kafka.client.keystore.jks -certreq -file client-cert-sign-request
    -alias Example-Alias -storepass Your-Store-Pass -keypass Your-Key-Pass
   ```
7. 打开 client-cert-sign-request 文件，并确保该文件的开头为 -----BEGIN CERTIFICATE REQUEST----- 且结尾为 -----END CERTIFICATE REQUEST-----。如果该

文件的开头为 -----BEGIN NEW CERTIFICATE REQUEST-----，请从文件的开头和结尾处删除单词 NEW（及其后面的单个空格）。

8. 在已 AWS CLI 安装证书的计算机上，运行以下命令对证书请求进行签名。将 *Private-CA-ARN* 替换为您的 PCA 的 ARN。如果需要，您可以更改有效性值。在这里，我们以 300 为例。

```
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"
```

保存响应中提供的证书 ARN。

> **Note**
>
> 要检索您的客户端证书，请使用 acm-pca get-certificate 命令并指定您的证书 ARN。有关更多信息，请参阅《AWS CLI Command Reference》中的 get-certificate。

9. 运行以下命令以获取为您 AWS 私有 CA 签名的证书。将 *Certificate-ARN* 替换为您从上一条命令的响应中获取的 ARN。

```
aws acm-pca get-certificate --certificate-authority-arn Private-CA-ARN --
certificate-arn Certificate-ARN
```

10. 从运行上一条命令所获得的 JSON 结果中，复制与 Certificate 和 CertificateChain 关联的字符串。将这两个字符串粘贴到名为的新文件中 signed-certificate-from-acm。先粘贴与 Certificate 关联的字符串，然后粘贴与 CertificateChain 关联的字符串。将 \n 字符替换为新行。以下是将证书和证书链粘贴到其中之后的文件结构。

```
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
```

11. 在客户端计算机上运行以下命令将此证书添加到您的密钥库中，以便能在与 MSK 代理交流时出示此证书。

```
keytool -keystore kafka.client.keystore.jks -import -file signed-certificate-from-
acm -alias Example-Alias -storepass Your-Store-Pass -keypass Your-Key-Pass
```

12. 使用以下内容创建名为 client.properties 的文件。将信任存储和密钥库位置调整为您将 kafka.client.truststore.jks 保存到的路径。用您的 Kafka 客户端版本替换 *{YOUR KAFKA VERSION}* 占位符。

```
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
```

使用身份验证生成和使用消息

1. 运行以下命令以创建主题。

```
<path-to-your-kafka-installation>/bin/kafka-topics.sh --create --
zookeeper ZooKeeper-Connection-String --replication-factor 3 --partitions 1 --topic
 ExampleTopic
```

2. 运行以下命令以启动控制台生成器。名为 client.properties 的文件是您在上一过程中创建的文件。

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

3. 在客户端计算机上的新命令窗口中，运行以下命令以启动控制台使用器。

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBroker-String --topic ExampleTopic --consumer.config
 client.properties
```

4. 在生成器窗口中键入消息，并观察消息显示在使用器窗口中。

使用 S AWS ecrets Manager 进行登录凭据身份验证

您可以使用使用 S AWS ecrets Manager 存储和保护的登录凭证来控制对您的 Amazon MSK 集群的访问权限。将用户凭证存储在 Secrets Manager 中可以减少集群身份验证的开销，例如审计、更新和轮换凭证。Secrets Manager 还让您能够跨集群共享用户凭证。

本主题包含下列部分：

工作方式

Amazon MSK 的登录凭证身份验证使用 SASL/SCRAM（Simple Authentication and Security Layer/Salted Challenge Response Mechanism）身份验证。要为集群设置登录凭证身份验证，您可以在 AWS Secrets Manager 中创建密钥资源，并将登录凭证与该密钥关联。

SASL/SCRAM 在 RFC 5802 中定义。SCRAM 使用安全哈希算法，不会在客户端和服务器之间传输明文登录凭证。

> **Note**
>
> 当您为集群设置 SASL/SCRAM 身份验证时，Amazon MSK 会为客户端和代理之间的所有流量开启 TLS 加密。

为 Amazon MSK 集群设置 SASL/SCRAM 身份验证

要在 Secr AWS ets Manager 中设置密钥，请按照 Secrets Man ager 用户指南中的创建和检索密AWS钥教程进行操作。

在为 Amazon MSK 集群创建密钥时，请注意以下要求：

- 对于密钥类型，请选择其他密钥类型（例如 API 密钥）。
- 您的密钥名称必须以前缀 AmazonMSK_ 开头。
- 您必须使用现有的自定义 AWS KMS 密钥或为您的密 AWS KMS 钥创建新的自定义密钥。默认情况下，Secrets Manager 对密 AWS KMS 钥使用默认密钥。

> **Important**
>
> 使用默认密钥创建的密 AWS KMS 钥不能用于 Amazon MSK 集群。

- 您的登录凭证数据必须采用以下格式，才能使用明文选项输入键值对。

```
{
  "username": "alice",
  "password": "alice-secret"
}
```

- 记录密钥的 ARN（Amazon 资源名称）值。
-

> **Important**
>
> 您不能将 Secrets Manager 密钥与超出 the section called “ 调整集群的大小：每个代理的分区数量” 中所述限制的集群关联。

- 如果您使用创建密钥，请为参数指定密钥 ID 或 ARN。 AWS CLI `kms-key-id`不要指定别名。
- 要将密钥与您的集群关联，请使用 Amazon MSK 控制台或 BatchAssociateScramSecret操作。

> **Important**
>
> 当您将密钥与集群关联时，Amazon MSK 会向该密钥附加资源策略，以允许您的集群访问和读取您定义的密钥值。您不应修改此资源策略。这样做可能会阻止您的集群访问密钥。

`BatchAssociateScramSecret` 操作的以下 JSON 输入示例将密钥与集群关联：

```
{
  "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"
  ]
}
```

使用登录凭证连接到集群

在创建密钥并将其与集群关联后，您便可以将客户端连接到集群。以下示例步骤演示如何将客户端连接到使用 SASL/SCRAM 身份验证的集群，以及如何通过示例主题生成和使用。

1. 使用以下命令检索集群详细信息。*ClusterArn*替换为集群的 Amazon 资源名称 (ARN)：

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

从命令的 JSON 结果中，保存与名为 ZookeeperConnectString 的字符串关联的值。

2. 要创建示例主题，在您的客户端计算机上运行以下命令。*ZookeeperConnectString*替换为您在上一步中录制的字符串。

```
<path-to-your-kafka-installation>/bin/kafka-topics.sh --create --
zookeeper ZookeeperConnectString --replication-factor 3 --partitions 1 --
topic ExampleTopicName
```

3. 在您的客户端计算机上，创建一个 JAAS 配置文件，其中包含存储在密钥中的用户凭证。例如，对于用户 alice，使用以下内容创建一个名为 users_jaas.conf 的文件。

```
KafkaClient {
   org.apache.kafka.common.security.scram.ScramLoginModule required
   username="alice"
   password="alice-secret";
};
```

4. 使用以下命令将 JAAS 配置文件导出为 KAFKA_OPTS 环境参数。

```
export KAFKA_OPTS=-Djava.security.auth.login.config=<path-to-jaas-file>/
users_jaas.conf
```

5. 在 ./tmp 目录中创建一个名为 kafka.client.truststore.jks 的文件。

6. 使用以下命令将 JDK 密钥存储文件从 JVM cacerts 文件夹复制到您在上一步中创建的 kafka.client.truststore.jks 文件。将 *JDKFolder* 替换为实例上 JDK 文件夹的名称。例如，您的 JDK 文件夹可能命名为 java-1.8.0-openjdk-1.8.0.201.b09-0.amzn2.x86_64。

```
cp /usr/lib/jvm/JDKFolder/jre/lib/security/cacerts /tmp/kafka.client.truststore.jks
```

7. 在 Apache Kafka 安装的 bin 目录中，创建一个名为 client_sasl.properties 的客户端属性文件，其中包含以下内容。此文件可定义 SASL 机制和协议。

```
security.protocol=SASL_SSL
sasl.mechanism=SCRAM-SHA-512
ssl.truststore.location=<path-to-keystore-file>/kafka.client.truststore.jks
```

8. 使用以下命令检索引导代理字符串。*ClusterArn*替换为集群的 Amazon 资源名称 (ARN)：

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

从命令的 JSON 结果中，保存与名为 BootstrapBrokerStringSaslScram 的字符串关联的值。

9. 要生成您创建的示例主题，请在客户端计算机上运行以下命令。*BootstrapBrokerStringSaslScram*替换为您在上一步中检索到的值。

```
<path-to-your-kafka-installation>/bin/kafka-console-producer.sh --broker-
list BootstrapBrokerStringSaslScram --topic ExampleTopicName --producer.config
 client_sasl.properties
```

10. 要使用您创建的主题，在您的客户端计算机上运行以下命令。*BootstrapBrokerStringSaslScram*用您之前获得的值替换。

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBrokerStringSaslScram --topic ExampleTopicName --from-beginning --
consumer.config client_sasl.properties
```

使用用户

创建用户：您在密钥中以键值对的形式创建用户。在 Secrets Manager 控制台中使用明文选项时，应按以下格式指定登录凭证数据。

```
{
  "username": "alice",
  "password": "alice-secret"
}
```

撤销用户访问权限：要撤销用户访问集群的凭证，建议您先在集群上移除或强制执行 ACL，然后取消与该密钥的关联。这是因为：

- 移除用户并不能关闭现有连接。
- 对密钥的更改最多需要 10 分钟才能传播。

有关将 ACL 与 Amazon MSK 结合使用的更多信息，请参阅 Apache Kafka ACL。

建议您限制对 zookeeper 节点的访问权限，以防止用户修改 ACL。有关更多信息，请参阅 控制对 Apache 的访问权限 ZooKeeper。

限制

使用 SCRAM 密钥时请注意以下限制：

- Amazon MSK 仅支持 SCRAM-SHA-512 身份验证。
- 一个 Amazon MSK 集群最多可拥有 1000 个用户。
- 你必须在你的密钥中 AWS KMS key 使用。您不能将使用默认 Secrets Manager 加密密钥的密钥与 Amazon MSK 一起使用。有关创建 KMS 密钥的信息，请参阅 Creating symmetric encryption KMS keys。
- 您无法在 Secrets Manager 中使用非对称 KMS 密钥。
- 使用该 BatchAssociateScramSecret操作，您一次最多可以将 10 个密钥与一个集群关联。
- 与 Amazon MSK 集群关联的密钥的名称必须带有前缀 AmazonMSK_。
- 与 Amazon MSK 集群关联的密钥必须与集群位于相同的 Amazon Web Services 账户和 AWS 区域中。

Apache Kafka ACL

Apache Kafka 有一个可插拔的授权器，并附带一个使用 Apache 存储所有 ACL 的 out-of-box 授权器实现。 ZooKeeper Amazon MSK 在代理上的 `server.properties` 文件中启用此授权方。对于 Apache Kafka 版本 2.4.1，授权方是。 AclAuthorizer对于早期版本的 Apache Kafka 来说，确实如此。SimpleAclAuthorizer

Apache Kafka ACL 的格式为 “主体 P 是 [允许/拒绝] 主机 H 对任何与 RP 匹配的资源 R 执行操作 O”。ResourcePattern 如果 RP 与特定资源 R 不匹配，则 R 没有关联的 ACL，因此不允许除超级用户之外的用户访问 R。若要更改此 Apache Kafka 行为，请将属性 `allow.everyone.if.no.acl.found` 设为 true。默认情况下，Amazon MSK 会将其设置为 true。这意味着，对于 Amazon MSK 集群，如果您没有在资源上显式设置 ACL，则所有委托人都可以访问此资源。如果在资源上启用 ACL，则只有授权的委托人才能访问它。如果要限制对主题的访问并使用 TLS 相互身份验证授权客户端，请使用

Apache Kafka 授权方 CLI 添加 ACL。有关添加、删除和列出 ACL 的更多信息，请参阅 Kafka 授权命令行界面。

除客户端之外，您还需要授予所有代理访问主题的权限，以便代理可以从主分区复制消息。如果代理无权访问某个主题，则该主题的复制将失败。

添加或删除对主题的读写访问权

1. 将代理添加到 ACL 表中，以允许它们读取具有 ACL 的所有主题。要授予代理对主题的读取访问权限，请在可与 MSK 集群通信的客户端计算机上运行以下命令。

   将*ZooKeeper########## Ap ZooKeeper ache #####*。有关如何获取此字符串的信息，请参阅the section called “获取 Apache ZooKeeper 连接字符串”。

   用任何集群引导代理的 DNS 替换 *Distinguished-Name*，然后用星号 (*) 替换此可分辨名称中第一个句点之前的字符串。例如，如果您的集群的引导代理之一具有 DNS `b-6.mytestcluster.67281x.c4.kafka.us-east-1.amazonaws.com`，请将以下命令中的 *Distinguished-Name* 替换为 `*.mytestcluster.67281x.c4.kafka.us-east-1.amazonaws.com`。有关如何获取引导代理的信息，请参阅the section called “获取引导代理”。

   ```
   <path-to-your-kafka-installation>/bin/kafka-acls.sh --authorizer-properties
    zookeeper.connect=ZooKeeper-Connection-String --add --allow-principal
    "User:CN=Distinguished-Name" --operation Read --group=* --topic Topic-Name
   ```

2. 要授予对主题的读访问权，请在客户端计算机上运行以下命令。如果使用双向 TLS 身份验证，请使用您创建私有密钥时使用的同一 *Distinguished-Name*。

   ```
   <path-to-your-kafka-installation>/bin/kafka-acls.sh --authorizer-properties
    zookeeper.connect=ZooKeeper-Connection-String --add --allow-principal
    "User:CN=Distinguished-Name" --operation Read --group=* --topic Topic-Name
   ```

   要删除读访问权，您可以运行相同的命令，并将 `--add` 替换为 `--remove`。

3. 要授予对主题的写访问权，请在客户端计算机上运行以下命令。如果使用双向 TLS 身份验证，请使用您创建私有密钥时使用的同一 *Distinguished-Name*。

   ```
   <path-to-your-kafka-installation>/bin/kafka-acls.sh --authorizer-properties
    zookeeper.connect=ZooKeeper-Connection-String --add --allow-principal
    "User:CN=Distinguished-Name" --operation Write --topic Topic-Name
   ```

要删除写访问权，您可以运行相同的命令，并将 `--add` 替换为 `--remove`。

更改 Amazon MSK 集群的安全组

本页介绍如何更改现有 MSK 集群的安全组。您可能需要更改集群的安全组，以便为特定用户组提供访问权限或限制对集群的访问权限。有关安全组的信息，请参阅《Amazon VPC 用户指南》中的您的 VPC 的安全组。

1. 使用中的 ListNodesAPI 或 list-nodes 命令获取集群中代理的列表。 AWS CLI 此操作的结果包括与代理关联的弹性网络接口（ENI）的 ID。
2. 登录 AWS Management Console 并打开亚马逊 EC2 控制台，网址为 https://console.aws.amazon.com/ec2/。
3. 使用屏幕右上角附近的下拉列表，选择部署集群的区域。
4. 在左侧窗格的网络与安全下，选择网络接口。
5. 选择您在第一步中获得的第一个 ENI。选择屏幕顶部的操作菜单，然后选择更改安全组。将新的安全组分配给此 ENI。对您在第一步中获得的每个 ENI 重复此步骤。

   > **Note**
   >
   > 您使用 Amazon EC2 控制台对集群安全组所做的更改，不会反映在 MSK 控制台的网络设置下。

6. 配置新安全组的规则，确保您的客户端可以访问代理。有关设置安全组规则的信息，请参阅《Amazon VPC 用户指南》中的添加、删除和更新规则。

> **Important**
>
> 如果您更改与集群代理关联的安全组，然后向该集群添加新的代理，Amazon MSK 会将新代理与创建集群时与该集群关联的原始安全组关联。但是，要使集群正常运行，其所有代理都必须与同一个安全组关联。因此，如果您在更改安全组后添加新代理，则必须再次执行前面的步骤并更新新代理的 ENI。

控制对 Apache 的访问权限 ZooKeeper

出于安全考虑，您可以限制对属于您的 Amazon MSK ZooKeeper 集群的 Apache 节点的访问。要限制对节点的访问，您可以为节点分配单独的安全组。然后，您可以决定有权访问该安全组的人员。

本主题包含下列部分：

- 将 Apache ZooKeeper 节点放在单独的安全组中
- 在 Apache 中使用 TLS 安全性 ZooKeeper

将 Apache ZooKeeper 节点放在单独的安全组中

1. 获取您的集群的 Apache ZooKeeper 连接字符串。要了解如何操作，请参阅 the section called “获取 Apache ZooKeeper 连接字符串”。连接字符串包含您的 Apache ZooKeeper 节点的 DNS 名称。
2. 使用 `host` 或 `ping` 等工具将您在上一步中获得的 DNS 名称转换为 IP 地址。稍后您需要在此过程中使用这些 IP 地址，因此请保存这些地址。
3. 登录 AWS Management Console 并打开亚马逊 EC2 控制台，网址为 https://console.aws.amazon.com/ec2/。
4. 在左侧窗格的 Network & Security (网络与安全性) 下，选择 Network Interfaces (网络接口)。
5. 在网络接口表上方的搜索字段中，键入集群名称，然后键入 return。这会将表中显示的网络接口数限制为与您的集群关联的接口。
6. 选中与列表中的第一个网络接口对应的行开头处的复选框。
7. 在页面底部的详细信息窗格中，查找 Primary private IPv4 IP (主要私有 IPv4 IP)。如果此 IP 地址与您在本过程第一步中获得的 IP 地址相匹配，则表示该网络接口已分配给集群中的一个 Apache ZooKeeper 节点。否则，取消选中此网络接口旁边的复选框，然后选择列表中的下一个网络接口。选择网络接口的顺序无关紧要。在接下来的步骤中，您将对分配给 Apache ZooKeeper 节点的所有网络接口逐一执行相同的操作。
8. 当您选择与 Apache ZooKeeper 节点对应的网络接口时，请选择页面顶部的操作菜单，然后选择更改安全组。将新安全组分配给此网络接口。有关创建安全组的信息，请参阅 Amazon VPC 文档中的创建安全组。
9. 重复上一步为与集群的 Apache ZooKeeper 节点关联的所有网络接口分配相同的新安全组。
10. 现在，您可以选择有权访问此新安全组的人员。有关设置安全组规则的信息，请参阅 Amazon VPC 文档中的添加、删除和更新规则。

在 Apache 中使用 TLS 安全性 ZooKeeper

您可以使用 TLS 安全性在客户端和 Apache ZooKeeper 节点之间传输时进行加密。要在 Apache ZooKeeper 节点上实现 TLS 安全，请执行以下操作：

- 集群必须使用 Apache Kafka 版本 2.5.1 或更高版本才能在 Apache 中使用 TLS 安全性。 ZooKeeper
- 在创建或配置集群时启用 TLS 安全。使用 Apache Kafka 版本 2.5.1 或更高版本创建并启用 TLS 的集群会自动对 Apache 终端节点使用 TLS 安全性。 ZooKeeper 有关设置 TLS 安全的信息，请参阅如何开始使用加密？。
- 使用DescribeCluster 操作检索 TLS Apache ZooKeeper 端点。
- 创建 Apache ZooKeeper 配置文件，以便与`kafka-configs.sh`和`kafka-acls.sh`工具或 ZooKeeper 外壳一起使用。对于每个工具，您都使用`--zk-tls-config-file`参数来指定 Apache ZooKeeper 配置。

  以下示例显示了一个典型的 Apache ZooKeeper 配置文件：

  ```
  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
  ```

- 对于其他命令（例如`kafka-topics`），必须使用`KAFKA_OPTS`环境变量来配置 Apache ZooKeeper 参数。以下示例说明如何配置`KAFKA_OPTS`环境变量以将 Apache ZooKeeper 参数传递给其他命令：

  ```
  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"
  ```

  配置 `KAFKA_OPTS` 环境变量后，您便可正常使用 CLI 命令。以下示例使用环境变量中的 Apache ZooKeeper 配置创建 Apache Kafka 主题：`KAFKA_OPTS`

```
<path-to-your-kafka-installation>/bin/kafka-topics.sh --create --
zookeeper ZooKeeperTLSConnectString --replication-factor 3 --partitions 1 --topic
 AWSKafkaTutorialTopic
```

> **Note**
>
> 您在 Apache ZooKeeper 配置文件中使用的参数名称与您在KAFKA_OPTS环境变量中使用的参数名称不一致。注意在配置文件和 KAFKA_OPTS 环境变量中与参数一起使用的名称。

有关使用 TLS 访问您的 Apache ZooKeeper 节点的更多信息，请参阅 KIP-515：启用 ZK 客户端使用新的 TLS 支持的身份验证。

日志记录

您可以将 Apache Kafka 代理日志传送到以下一种或多种目标类型：亚马逊日 CloudWatch 志、亚马逊 S3、Amazon Data Firehose。您也可以使用记录亚马逊 MSK API 调用。 AWS CloudTrail

代理日志

利用代理日志，您可以对 Apache Kafka 应用程序进行问题排查，并分析它们与 MSK 集群的通信。您可以将新的或现有 MSK 集群配置为将信息级代理日志传送到以下一种或多种目标资源： CloudWatch 日志组、S3 存储桶、Firehose 传输流。然后，您可以通过 Firehose 将传输流中的日志数据传送到 OpenSearch 服务。在配置集群以向其传送代理日志之前，必须创建目标资源。如果尚不存在这些目标资源，Amazon MSK 也不会为您创建。有关这三种类型的目标资源以及如何创建这些资源的信息，请参阅以下文档：

- Amazon CloudWatch 日志
- Amazon S3
- Amazon Data Firehose

所需的权限

要为 Amazon MSK 代理日志配置目标，您用于 Amazon MSK 操作的 IAM 身份必须具有 AWS 托管策略：AmazonMSK FullAccess 策略中所述的权限。

要将代理日志流式传输到 S3 存储桶，您还需要 `s3:PutBucketPolicy` 权限。有关 S3 存储桶策略的信息，请参阅《Amazon S3 用户指南》中的如何添加 S3 存储桶策略？。有关 IAM 策略的一般信息，请参阅《IAM 用户指南》中的访问管理。

与 SSE-KMS 存储桶结合使用时必需的 KMS 密钥政策

如果您使用带有客户托管密钥的 AWS KMS托管密钥 (SSE-KMS) 为 S3 存储桶启用了服务器端加密，请将以下内容添加到您的 KMS 密钥的密钥策略中，以便 Amazon MSK 可以将代理文件写入存储桶。

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

使用配置代理日志 AWS Management Console

如果您要创建新集群，请在监控部分中查找代理日志传送标题。您可以指定希望 Amazon MSK 向其传送代理日志的目标。

对于现有集群，请从集群列表中选择集群，然后选择属性选项卡。向下滚动到日志传送部分，然后选择其编辑按钮。您可以指定希望 Amazon MSK 向其传送代理日志的目标。

使用配置代理日志 AWS CLI

使用 `create-cluster` 或 `update-monitoring` 命令时，您可以选择指定 `logging-info` 参数并将类似如下的 JSON 结构传递给该参数。在此 JSON 中，所有三种目标类型都是可选的。

```
{
```

```
  "BrokerLogs": {
    "S3": {
      "Bucket": "ExampleBucketName",
      "Prefix": "ExamplePrefix",
      "Enabled": true
    },
    "Firehose": {
      "DeliveryStream": "ExampleDeliveryStreamName",
      "Enabled": true
    },
    "CloudWatchLogs": {
      "Enabled": true,
      "LogGroup": "ExampleLogGroupName"
    }
  }
}
```

使用 API 配置代理日志

您可以在 JSON 中指定传递给CreateCluster或UpdateMonitoring操作的可选`loggingInfo`结构。

> **Note**
>
> 默认情况下，启用代理日志记录后，Amazon MSK 会将 `INFO` 级别日志记录到指定目标。但是，Apache Kafka 2.4.X 及更高版本的用户可以将代理日志级别动态设置为任何 log4j 日志级别。有关动态设置代理日志级别的信息，请参阅 KIP-412: Extend Admin API to support dynamic application log levels。如果您将日志级别动态设置为`DEBUG`或`TRACE`，我们建议使用 Amazon S3 或 Firehose 作为日志目标。如果您使用 CloudWatch 日志作为日志目标，并且动态启用`DEBUG`或`TRACE`级别日志记录，Amazon MSK 可能会持续提供日志样本。这可能会对代理性能带来显著影响，因此只有在 `INFO` 日志级别不够详细，无法确定问题的根本原因时才应使用。

使用 记录 AWS CloudTrail API 调用

> **Note**
>
> AWS CloudTrail 只有在您使用IAM 访问控制时，日志才可用于 Amazon MSK。

Amazon MSK 与 AWS CloudTrail一项服务集成，该服务提供用户、角色或 AWS 服务在 Amazon MSK 中采取的操作的记录。 CloudTrail 将发出的 API 调用捕获为事件。捕获的调用包含来自 Amazon MSK 控制台的调用以及对 Amazon MSK API 操作的代码调用。它还会捕获 Apache Kafka 操作，例如创建和更改主题与组。

如果您创建了跟踪，则可以允许将 CloudTrail 事件持续传输到 Amazon S3 存储桶，包括 Amazon MSK 的事件。如果您未配置跟踪，您仍然可以在 CloudTrail 控制台的 “事件历史记录” 中查看最新的事件。使用收集的信息 CloudTrail，您可以确定向 Amazon MSK 或 Apache Kafka 操作发出的请求、发出请求的 IP 地址、谁发出了请求、何时发出请求以及其他详细信息。

要了解更多信息 CloudTrail，包括如何配置和启用它，请参阅AWS CloudTrail 用户指南。

亚马逊 MSK 信息位于 CloudTrail

CloudTrail 在您创建账户时，您的亚马逊 Web Services 账户已启用。当 MSK 集群中出现支持的事件活动时，该活动会与其他 AWS 服务 CloudTrail 事件一起记录在事件历史记录中。您可以在 Amazon Web Services 账户中查看、搜索和下载最新事件。有关更多信息，请参阅使用事件历史记录查看 CloudTrail事件。

要持续记录 Amazon Web Services 账户中的事件（包括 Amazon MSK 的事件），请创建跟踪。跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。预设情况下，在控制台中创建跟踪时，此跟踪应用于所有 区域。此跟踪记录在 AWS 分区中记录所有区域中的事件，并将日志文件传送至您指定的 Amazon S3 存储桶。此外，您可以配置其他 Amazon 服务，以进一步分析和处理 CloudTrail 日志中收集的事件数据。有关更多信息，请参阅下列内容：

- 创建跟踪概述
- CloudTrail 支持的服务和集成
- 配置 Amazon SNS 通知 CloudTrail
- 接收来自多个区域的 CloudTrail 日志文件和接收来自多个账户的 CloudTrail 日志文件

Amazon MSK 将所有 Amazon MSK 操作作为事件 CloudTrail 记录在日志文件中。此外，它还会记录以下 Apache Kafka 操作。

- kafka 集群：DescribeClusterDynamicConfiguration
- kafka 集群：AlterClusterDynamicConfiguration
- kafka 集群：CreateTopic
- kafka 集群：DescribeTopicDynamicConfiguration
- kafka 集群：AlterTopic

- kafka 集群：AlterTopicDynamicConfiguration
- kafka 集群：DeleteTopic

每个事件或日记账条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根用户还是 AWS Identity and Access Management (IAM) 用户证书发出。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其他 AWS 服务发出。

有关更多信息，请参阅CloudTrail 用户身份元素。

示例：Amazon MSK 日志文件条目

跟踪是一种配置，允许将事件作为日志文件传输到您指定的 Amazon S3 存储桶。 CloudTrail 日志文件包含一个或多个日志条目。一个事件表示来自任何源的一个请求，包括有关所请求的操作、操作的日期和时间、请求参数等方面的信息。 CloudTrail 日志文件不是公共 API 调用和 Apache Kafka 操作的有序堆栈跟踪，因此它们不会按任何特定的顺序出现。

以下示例显示了演示DescribeCluster和 DeleteCluster Amazon MSK 操作的 CloudTrail 日志条目。


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

以下示例显示了演示该kafka-cluster:CreateTopic操作的 CloudTrail 日志条目。

```
{
  "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"
}
```

Amazon Managed Streaming for Apache Kafka 的合规性验证

作为 AWS 合规性计划的一部分，第三方审计员将评估 Amazon Managed Streaming for Apache Kafka 的安全性与合规性。其中包括 PCI 和 HIPAA BAA。

有关特定合规计划范围内的 AWS 服务列表，请参阅按合规计划提供的范围内的亚马逊服务 Amazon Web Ser 。有关一般信息，请参阅AWS 合规计划AWS。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的 “下载报告” 中的 “ AWS Artifact。

您在使用 Amazon MSK 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。 AWS 提供了以下资源来帮助实现合规性：

- 安全性与合规性快速入门指南 - 这些部署指南讨论了架构注意事项，并提供了在 AWS上部署基于安全性和合规性的基准环境的步骤。
- HIPAA 安全与合规架构白皮书 — 本白皮书描述了公司如何使用来 AWS 创建符合 HIPAA 标准的应用程序。
- AWS 合规资源AWS — 此工作簿和指南集可能适用于您所在的行业和所在地区。
- 使用AWS Config 开发人员指南中的规则评估资源 — 该 AWS Config 服务评估您的资源配置在多大程度上符合内部实践、行业准则和法规。
- AWS Security Hub— 此 AWS 服务可全面了解您的安全状态 AWS ，帮助您检查是否符合安全行业标准和最佳实践。

Amazon Managed Streaming for Apache Kafka 中的恢复能力

AWS 全球基础设施是围绕 AWS 区域和可用区构建的。 AWS 区域提供多个物理隔离和隔离的可用区，这些可用区通过低延迟、高吞吐量和高度冗余的网络相连。利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错性和可扩展性。

有关 AWS 区域和可用区的更多信息，请参阅AWS 全球基础设施。

Amazon Managed Streaming for Apache Kafka 中的基础设施安全性

作为一项托管服务，适用于 Apache Kafka 的亚马逊托管流媒体受 AWS 亚马逊网络服务：安全流程概述白皮书中描述的全球网络安全程序的保护。

您可以使用 AWS 已发布的 API 调用通过网络访问 Amazon MSK。客户端必须支持传输层安全性协议 (TLS) 1.0 或更高版本。建议使用 TLS 1.2 或更高版本。客户端还必须支持具有完全向前保密（PFS）的密码套件，例如 Ephemeral Diffie-Hellman（DHE）或 Elliptic Curve Ephemeral Diffie-Hellman（ECDHE）。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

此外，必须使用访问密钥 ID 和与 IAM 委托人关联的秘密访问密钥来对请求进行签名。或者，您可以使用 AWS Security Token Service（AWS STS）生成临时安全凭证来对请求进行签名。

连接到 Amazon MSK 集群

默认情况下，只有当客户端与 MSK 集群位于同一 VPC 中时，前者才能访问后者。默认情况下，Kafka 客户端和 MSK 集群之间的所有通信都是私密的，您的流数据永远不会通过互联网传输。要从与 MSK 集群位于同一 VPC 中的客户端连接到该集群，请确保集群的安全组具有接受来自客户端安全组流量的入站规则。有关设置这些规则的信息，请参阅安全组规则。有关如何从与集群位于同一 VPC 中的 Amazon EC2 实例访问集群的示例，请参阅 开始使用。

要从集群 VPC 之外的客户端连接到您的 MSK 集群，请参阅从集群的 VPC 内部 AWS 但外部进行访问。

主题

公有访问权限

Amazon MSK 允许您选择开启对运行 Apache Kafka 2.6.0 或更高版本的 MSK 集群代理的公共访问权限。出于安全考虑，您在创建 MSK 集群时无法开启公共访问权限。但是，您可以更新现有集群以使其可供公开访问。您还可以创建一个新集群，然后对其进行更新，使其可供公开访问。

您可以开启对 MSK 集群的公共访问权限，无需支付额外费用，但是进出集群的数据需要支付标准 AWS 的数据传输费用。有关定价的信息，请参阅 Amazon EC2 按需定价。

要开启对集群的公共访问权限，请先确保集群满足以下所有条件：

- 与集群关联的子网必须是公有子网。这意味着子网必须具有关联的路由表并连接了互联网网关。有关如何创建和附加互联网网关的信息，请参阅《Amazon VPC 用户指南》中的互联网网关。
- 未经身份验证的访问控制必须关闭，并且必须至少开启以下访问控制方法之一：SASL/IAM、SASL/SCRAM、mTLS。有关如何更新集群的访问控制方法的信息，请参阅 the section called “更新安全设置”。
- 必须开启集群内的加密。开启设置是创建集群时的默认设置。对于在集群中的加密处于关闭状态时创建的集群，无法为其开启加密。因此，对于在集群中的加密处于关闭状态时创建的集群，无法为其开启公共访问权限。
- 代理和客户端之间的明文流量必须关闭。有关在其开启时如何关闭的信息，请参阅 the section called “更新安全设置”。

- 如果您使用的是 SASL/SCRAM 或 mTLS 访问控制方法，则必须为集群设置 Apache Kafka ACL。为集群设置 Apache Kafka ACL 后，更新集群的配置，以将该集群的属性 `allow.everyone.if.no.acl.found` 设置为 false。有关如何更新集群配置的信息，请参阅 the section called “配置操作”。如果您使用的是 IAM 访问控制并想要应用授权策略或更新授权策略，请参阅 the section called “IAM 访问控制”。有关 Apache Kafka ACL 的信息，请参阅 the section called “Apache Kafka ACL”。

在您确保 MSK 集群满足上面列出的条件后，您可以使用 AWS Management Console AWS CLI、或 Amazon MSK API 开启公共访问权限。开启集群的公共访问权限后，您可以为其获取一个公共引导代理字符串。有关获取集群引导代理的信息，请参阅 the section called “获取引导代理”。

Important

除了开启公共访问权限外，还要确保集群的安全组具有允许从您的 IP 地址进行公共访问的入站 TCP 规则。因此，建议您尽可能严格设置这些规则。有关安全组和入站规则的信息，请参阅《Amazon VPC 用户指南》中的 您的 VPC 的安全组。有关端口号，请参阅 the section called “端口信息”。有关如何更改集群安全组的说明，请参阅 the section called “更改安全组”。

Note

如果您按照以下说明开启公共访问权限，但仍无法访问集群，请参阅 the section called “无法访问已开启公共访问权限的集群”。

使用控制台开启公共访问权限

1. 登录并打开亚马逊 MSK 控制台，网址为 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。 AWS Management Console
2. 在集群列表中，选择要为其开启公共访问权限的集群。
3. 选择属性选项卡，然后找到网络设置部分。
4. 选择编辑公共访问权限。

使用开启公共访问权限 AWS CLI

1. 运行以下 AWS CLI 命令，将 *"######" ##ClusterArn# ARN ###*的当前版本。要查找集群的当前版本，请使用DescribeCluster操作或 desc ribe- AWS CLI cluster 命令。示例版本是 KTVPDKIKX0DER。

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

该 update-connectivity 命令的输出如以下 JSON 示例所示。

```
{
    "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**
>
> 要关闭公共访问权限，请使用类似的 AWS CLI 命令，但改为使用以下连接信息：
>
> ```
> '{"PublicAccess": {"Type": "DISABLED"}}'
> ```

2. 要获得update-connectivity操作结果，请运行以下命令，*ClusterOperationArn*替换为在命令输出中获得的 ARN。update-connectivity

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

该 describe-cluster-operation 命令的输出如以下 JSON 示例所示。

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

如果 `OperationState` 的值为 `UPDATE_IN_PROGRESS`，请等待一段时间，然后再次运行 `describe-cluster-operation` 命令。

使用 Amazon MSK API 开启公共访问权限

- 要使用 API 开启或关闭集群的公共访问权限，请参阅UpdateConnectivity。

> **Note**
>
> 出于安全考虑，Amazon MSK 不允许公众访问 Apache 节点 ZooKeeper 。有关如何从内部控制对 MSK 集群的 Apache ZooKeeper 节点的访问的信息 AWS，请参阅。the section called “控制对 Apache 的访问权限 ZooKeeper”

从集群的 VPC 内部 AWS 但外部进行访问

要从集群的 Amazon VPC 内部 AWS 但外部连接到 MSK 集群，可以使用以下选项。

Amazon VPC 对等连接

要从不同于集群 VPC 的 VPC 连接到 MSK 集群，您可以在这两个 VPC 之间建立对等连接。有关 VPC 对等连接的信息，请参阅 Amazon VPC 对等连接指南。

AWS Direct Connect

AWS Direct Connect 通过标准的 1 千兆位或 10 千兆位以太网光纤电缆将 AWS 您的本地网络链接到该网络。电缆的一端连接到您的路由器，另一端连接到 AWS Direct Connect 路由器。建立此连接后，您可以直接创建通往 AWS 云和 Amazon VPC 的虚拟接口，绕过网络路径中的互联网服务提供商。有关更多信息，请参阅 AWS Direct Connect。

AWS Transit Gateway

AWS Transit Gateway 是一项服务，可让您将 VPC 和本地网络连接到单个网关。有关如何使用 AWS Transit Gateway的信息，请参阅AWS Transit Gateway。

VPN 连接

您可以使用以下主题中介绍的 VPN 连接选项，将 MSK 集群的 VPC 连接到远程网络和用户：VPN 连接。

REST 代理

您可以在集群的 Amazon VPC 中运行的实例上安装 REST 代理。利用 REST 代理，生成器和使用器能够通过 HTTP API 请求与集群通信。

多区域多 VPC 连接

以下文档介绍了位于不同区域的多个 VPC 的连接选项：多区域多 VPC 连接。

单区域多 VPC 私有连接

适用于 Apache Managed Streaming Kafka（Amazon MSK）集群的多 VPC 私有连接（由 AWS PrivateLink）提供支持，该功能使您能够更快地将托管在不同虚拟私有云 (VPC) AWS 和账户中的 Kafka 客户端连接到亚马逊 MSK 集群。

请参阅 Single Region multi-VPC connectivity for cross-account clients。

EC2-Classic 网络已停用

亚马逊 MSK 不再支持使用亚马逊 EC2-Classic 网络运行的亚马逊 EC2 实例。

请参阅 EC2-Classic 网络即将停用——以下是准备方法。

单区域中的 Amazon MSK 多 VPC 私有连接

适用于 Apache Managed Streaming Kafka（Amazon MSK）集群的多 VPC 私有连接（由 AWS PrivateLink）提供支持，该功能使您能够更快地将托管在不同虚拟私有云 (VPC) AWS 和账户中的 Kafka 客户端连接到亚马逊 MSK 集群。

多 VPC 私有连接是一种托管式解决方案，可简化多 VPC 和跨账户连接的网络基础设施。客户端可以通过连接到 Amazon MSK 集群， PrivateLink 同时将所有流量保持在 AWS 网络内。适用于亚马逊 MSK 集群的多 VPC 私有连接适用于所有可用 Amazon MSK 的 AWS 区域。

主题

什么是多 VPC 私有连接？

Amazon MSK 的多 VPC 私有连接是一种连接选项，允许您将托管在不同虚拟私有云 (VPC) 和 AWS 账户中的 Apache Kafka 客户端连接到 MSK 集群。

Amazon MSK 通过集群策略简化跨账户存取。这些策略允许集群所有者向其他 AWS 账户授予与 MSK 集群建立私有连接的权限。

多 VPC 私有连接的优势

与其他连接解决方案相比，多 VPC 私有连接具有以下几个优势：

- 它可以自动执行 AWS PrivateLink 连接解决方案的运营管理。
- 它允许在连接的 VPC 之间重叠 IP，从而无需维护与其他 VPC 连接解决方案关联的非重叠的 IP、复杂的对等连接和路由表。

您可以使用适用于 MSK 集群的集群策略来定义哪些 AWS 账户有权设置与 MSK 集群的跨账户私有连接。跨账户管理员可以将权限委派给相应的角色或用户。当与 IAM 客户端身份验证一起使用时，您也可以使用集群策略为连接的客户端精细定义 Kafka 数据面板的权限。

多 VPC 私有连接的要求和限制

请注意运行多 VPC 私有连接的以下 MSK 集群要求：

- 只有 Apache Kafka 2.7.1 或更高版本支持多 VPC 私有连接。请确保与 MSK 集群搭配使用的任何客户端都运行与集群兼容的 Apache Kafka 版本。
- 多 VPC 私有连接支持身份验证类型 IAM、TLS 和 SASL/SCRAM。未经身份验证的集群无法使用多 VPC 私有连接。
- 如果您使用的是 SASL/SCRAM 或 mTLS 访问控制方法，则必须为集群设置 Apache Kafka ACL。首先，为集群设置 Apache Kafka ACL。然后，更新集群的配置，将集群的属性 `allow.everyone.if.no.acl.found` 设置为 false。有关如何更新集群配置的信息，请参阅 the section called “配置操作”。如果您使用的是 IAM 访问控制并想要应用授权策略或更新授权策略，请参阅 the section called “IAM 访问控制”。有关 Apache Kafka ACL 的信息，请参阅 the section called “Apache Kafka ACL”。
- 多 VPC 私有连接不支持 t3.small 实例类型。
- 不支持跨 AWS 区域的多 VPC 私有连接，仅支持同一区域内的 AWS 账户。
- Amazon MSK 不支持与 Zookeeper 节点的多 VPC 私有连接。

开始使用多 VPC 私有连接

主题

- 步骤 1：在账户 A 的 MSK 集群上，为集群上的 IAM 身份验证方案开启多 VPC 连接
- 步骤 2：将集群策略附加到 MSK 集群
- 步骤 3：用于配置客户端托管的 VPC 连接的跨账户用户操作

本教程使用一个常见的用例作为示例，说明如何使用多 VPC 连接将 Apache Kafka 客户端从集群的 VPC 内部 AWS但外部私有地连接到 MSK 集群。此过程要求跨账户用户为每个客户端创建 MSK 托

管式 VPC 连接和配置，包括所需的客户端权限。该过程还要求 MSK 集群所有者在 MSK 集群上启用 PrivateLink 连接，并选择身份验证方案来控制对集群的访问。

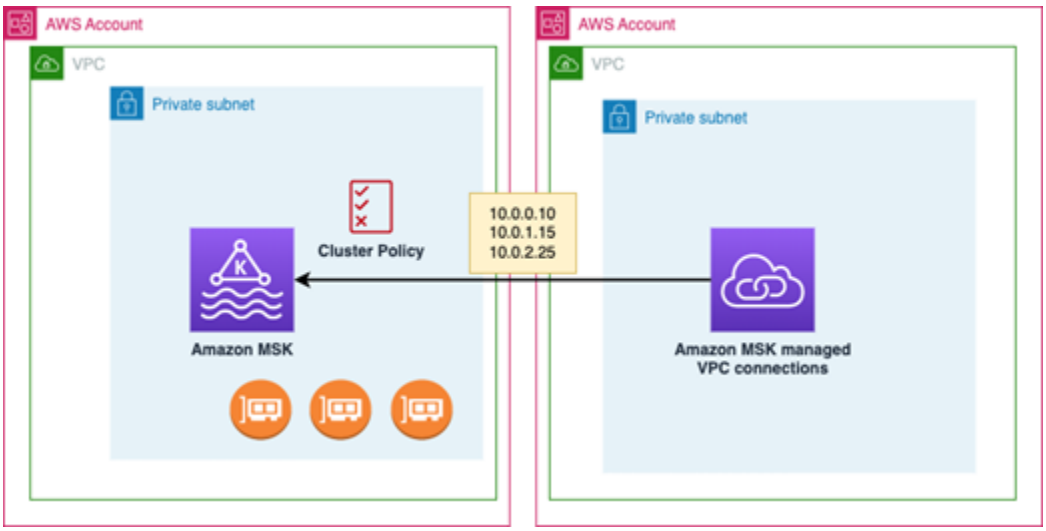
在本教程的不同部分中，我们选择适用于此示例的选项。这并不意味着它们是可用于设置 MSK 集群或客户端实例的唯一选项。

此用例的网络配置如下：

- 跨账户用户（Kafka 客户端）和 MSK 集群位于同一个 AWS 网络/区域中，但在不同的账户中：
  - 账户 A 中的 MSK 集群
  - 账户 B 中的 Kafka 客户端
- 跨账户用户将使用 IAM 身份验证方案私密连接到 MSK 集群。

本教程假设有一个使用 Apache Kafka 版本 2.7.1 或更高版本创建的预置 MSK 集群。在开始配置过程之前，MSK 集群必须处于 ACTIVE 状态。为避免潜在的数据丢失或停机，将使用多 VPC 私有连接来连接到集群的客户端应使用与集群兼容的 Apache Kafka 版本。

下图说明了连接到不同 AWS 账户中的客户端的 Amazon MSK 多 VPC 连接架构。



步骤 1：在账户 A 的 MSK 集群上，为集群上的 IAM 身份验证方案开启多 VPC 连接

MSK 集群所有者需要在 MSK 集群创建并处于 ACTIVE 状态后在该集群上进行配置设置。

集群所有者需为将在集群上处于活动状态的任何身份验证方案，在处于 ACTIVE 状态的集群上开启多 VPC 私有连接。这可以使用 UpdateSecurity API 或 MSK 控制台来完成。IAM、SASL/SCRAM 和 TLS 身份验证方案支持多 VPC 私有连接。无法为未经身份验证的集群启用多 VPC 私有连接。

对于此用例，您需要将集群配置为使用 IAM 身份验证方案。

> **Note**
>
> 如果您将 MSK 集群配置为使用 SASL/SCRAM 身份验证方案，则必须提供 Apache Kafka ACL 属性“allow.everyone.if.no.acl.found=false”。请参阅 Apache Kafka ACL。

当您更新多 VPC 私有连接设置时，Amazon MSK 会启动代理节点滚动重启，以更新代理配置。完成此过程可能最多需要 30 分钟或更长时间。在更新连接时，您无法对集群进行其他更新。

使用控制台为账户 A 中的集群上的选定身份验证方案开启多 VPC

1. 通过以下网址为集群所在的账户打开 Amazon MSK 控制台：https://console.aws.amazon.com/msk/。
2. 在导航窗格的 MSK 集群下，选择集群以显示账户中的集群列表。
3. 选择要为多 VPC 私有连接配置的集群。集群必须处于 ACTIVE 状态。
4. 选择集群属性选项卡，然后转到网络设置。
5. 选择编辑下拉菜单，然后选择开启多 VPC 连接。
6. 选择要为此集群开启的一种或多种身份验证类型。对于此用例，请选择基于 IAM 角色的身份验证。
7. 选择保存更改。

Example -在 UpdateConnectivity 集群上开启多 VPC 私有连接身份验证方案的 API

作为 MSK 控制台的替代方案，您可以使用 UpdateConnectivity API 开启多 VPC 私有连接，并在活动集群上配置身份验证方案。以下示例显示为集群开启了 IAM 身份验证方案。

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

```
  }
}
```

Amazon MSK 可创建私有连接所需的网络基础设施。Amazon MSK 还可为需要私有连接的每种身份验证类型创建一组新的引导代理端点。请注意，明文身份验证方案不支持多 VPC 私有连接。

步骤 2：将集群策略附加到 MSK 集群

集群所有者可以将集群策略（也称为基于资源的策略）附加到 MSK 集群，您将在其中开启多 VPC 私有连接。集群策略会授予客户端从其他账户访问集群的权限。在编辑集群策略之前，您需要应有权访问 MSK 集群的账户的账户 ID。请参阅 How Amazon MSK works with IAM。

集群所有者必须将集群策略附加到 MSK 集群，该策略将授权账户 B 中的跨账户用户获取集群的引导代理，并授权对账户 A 中的 MSK 集群执行以下操作：

- CreateVpcConnection
- GetBootstrapBrokers
- DescribeCluster
- DescribeClusterV2

Example

作为参考，以下是基本集群策略的 JSON 示例，类似于 MSK 控制台 IAM policy 编辑器中显示的默认策略。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            },
            "Action": [
                "kafka:CreateVpcConnection",
                "kafka:GetBootstrapBrokers",
                "kafka:DescribeCluster",
                "kafka:DescribeClusterV2"
```

```
            ],
            "Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/testing/
de8982fa-8222-4e87-8b20-9bf3cdfa1521-2"
        }
    ]
}
```

将集群策略附加到 MSK 集群

1. 在 Amazon MSK 控制台的 MSK 集群下，选择集群。
2. 向下滚动到安全设置，然后选择编辑集群策略。
3. 在控制台的编辑集群策略屏幕上，选择多 VPC 连接的基本策略。
4. 在账户 ID 字段中，输入应有权访问此集群的每个账户的账户 ID。在您输入 ID 时，它会自动复制到显示的策略 JSON 语法中。在我们的示例集群策略中，账户 ID 为 123456789012。
5. 选择保存更改。

有关集群策略 API 的信息，请参阅 Amazon MSK resource-based policies。

步骤 3：用于配置客户端托管的 VPC 连接的跨账户用户操作

要在与 MSK 集群不同的账户中的客户端之间设置多 VPC 私有连接，跨账户用户需要为该客户端创建托管式 VPC 连接。重复此程序，即可将多个客户端连接到 MSK 集群。在本用例中，您只需要配置一个客户端。

客户端可以使用支持的身份验证方案 IAM、SASL/SCRAM 或 TLS。每个托管式 VPC 连接只能与一个身份验证方案关联。必须在客户端将要连接的 MSK 集群上配置客户端身份验证方案。

对于此用例，请配置客户端身份验证方案，以便账户 B 中的客户端使用 IAM 身份验证方案。

先决条件

此过程需要以下项目：

- 先前创建的集群策略，可向账户 B 中的客户端授予对账户 A 中的 MSK 集群执行操作的权限。
- 附加到账户 B 中客户的身份策略，用于授予kafka:CreateVpcConnectionec2:CreateTags、ec2:CreateVPCEndpoint和ec2:DescribeVp作的权限。

Example

以下是基本客户端身份策略的 JSON 示例，供您参考。

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

为账户 B 中的客户端创建托管式 VPC 连接

1. 从集群管理员处获取您希望账户 B 中的客户端连接到的账户 A 中 MSK 集群的集群 ARN。记下集群 ARN 以供将来使用。
2. 在客户端账户 B 的 MSK 控制台中，选择托管式 VPC 连接，然后选择创建连接。
3. 在连接设置窗格中，将集群 ARN 粘贴到集群 ARN 文本字段中，然后选择验证。
4. 在账户 B 中选择客户端的身份验证类型。对于此用例，请在创建客户端 VPC 连接时选择 IAM。
5. 为客户端选择 VPC。
6. 至少选择两个可用区和关联的子网。您可以从 AWS 管理控制台集群详细信息中获取可用区 ID，也可以使用 DescribeClusterAPI 或 desc ribe-cluster AWS CLI 命令获取。您为客户端子网指定的区域 ID 必须与集群子网的区域 ID 相匹配。如果缺少子网的值，请先创建一个与 MSK 集群具有相同区域 ID 的子网。
7. 为此 VPC 连接选择安全组。您可以使用默认安全组。有关配置安全组的更多信息，请参阅 Control traffic to resources using security groups。
8. 选择创建连接。
9. 要从跨账户用户的 MSK 控制台（集群详细信息 > 托管式 VPC 连接）获取新引导代理字符串的列表，请参阅集群连接字符串下显示的引导代理字符串。在客户账户 B 中，可以通过调用 GetBootstrapBrokersAPI 或在控制台集群详细信息中查看引导代理列表来查看引导代理列表。

10. 按以下步骤更新与 VPC 连接关联的安全组：

    a. 为 PrivateLink VPC 设置入站规则，以允许来自账户 B 网络的 IP 范围的所有流量。

    b. [可选] 设置与 MSK 集群的出站规则连接。在 VPC 控制台中依次选择安全组、编辑出站规则，然后为端口范围 14001-14100 添加自定义 TCP 流量的规则。多 VPC 网络负载均衡器正在监听 14001-14100 端口范围。请参阅网络负载均衡器。

11. 将账户 B 中的客户端配置为使用用于多 VPC 私有连接的新引导代理连接到账户 A 中的 MSK 集群。请参阅 Produce and consume data。

授权完成后，Amazon MSK 会为每个指定的 VPC 和身份验证方案创建托管式 VPC 连接。所选安全组与每个连接相关联。此托管式 VPC 连接由 Amazon MSK 配置为私密地连接到代理。您可以使用一组新的引导代理私密地连接到 Amazon MSK 集群。

更新集群上的授权方案

多 VPC 私有连接支持多种授权方案：SASL/SCRAM、IAM 和 TLS。集群所有者可以为一个或多个身份验证方案开启/关闭私有连接。集群必须处于 ACTIVE 状态才能执行此操作。

使用 Amazon MSK 控制台开启身份验证方案

1. 在 AWS Management Console 中为要编辑的集群打开 Amazon MSK 控制台。
2. 在导航窗格的 MSK 集群下，选择集群以显示账户中的集群列表。
3. 选择要编辑的集群。集群必须处于 ACTIVE 状态。
4. 选择集群属性选项卡，然后转到网络设置。
5. 选择编辑下拉菜单，然后选择开启多 VPC 连接，以开启新的身份验证方案。
6. 选择要为此集群开启的一种或多种身份验证类型。
7. 选择开启选择。

当您开启新的身份验证方案时，您还应该为新的身份验证方案创建新的托管式 VPC 连接，并更新客户端，以使用特定于新身份验证方案的引导代理。

使用 Amazon MSK 控制台关闭身份验证方案

> **Note**
>
> 当您为身份验证方案关闭多 VPC 私有连接时，所有与连接相关的基础设施，包括托管式 VPC 连接，都将被删除。

当您为身份验证方案关闭多 VPC 私有连接时，客户端的现有 VPC 连接将变为 INACTIVE 状态，集群端的 Privatelink 基础设施（包括托管式 VPC 连接）将被删除。跨账户用户只能删除处于非活动状态的 VPC 连接。如果在集群上再次开启私有连接，则跨账户用户需要创建与集群的新连接。

1. 在 AWS Management Console 打开 Amazon MSK 控制台。
2. 在导航窗格的 MSK 集群下，选择集群以显示账户中的集群列表。
3. 选择要编辑的集群。集群必须处于 ACTIVE 状态。
4. 选择集群属性选项卡，然后转到网络设置。
5. 选择编辑下拉菜单，然后选择关闭多 VPC 连接，以关闭身份验证方案。
6. 选择要为此集群关闭的一种或多种身份验证类型。
7. 选择关闭选择。

Example 使用 API 开启/关闭身份验证方案

作为 MSK 控制台的替代方案，您可以使用 UpdateConnectivity API 开启多 VPC 私有连接，并在活动集群上配置身份验证方案。以下示例显示为集群开启了 SASL/SCRAM 和 IAM 身份验证方案。

当您开启新的身份验证方案时，您还应该为新的身份验证方案创建新的托管式 VPC 连接，并更新客户端，以使用特定于新身份验证方案的引导代理。

当您为身份验证方案关闭多 VPC 私有连接时，客户端的现有 VPC 连接将变为 INACTIVE 状态，集群端的 Privatelink 基础设施（包括托管式 VPC 连接）将被删除。跨账户用户只能删除处于非活动状态的 VPC 连接。如果在集群上再次开启私有连接，则跨账户用户需要创建与集群的新连接。

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

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

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

拒绝与 Amazon MSK 集群建立托管式 VPC 连接

通过集群管理员账户的 Amazon MSK 控制台，您可以拒绝客户端 VPC 连接。客户端 VPC 连接必须处于可用状态才能被拒绝。您可能需要拒绝来自不再有权连接到集群的客户端的托管式 VPC 连接。要防止新的托管式 VPC 连接连接到客户端，请在集群策略中拒绝对客户端的访问。在连接所有者删除被拒绝的连接之前，该连接仍会产生费用。请参阅 Delete a managed VPC connection to an Amazon MSK cluster。

使用 MSK 控制台拒绝客户端 VPC 连接

1. 在 AWS Management Console 打开 Amazon MSK 控制台。
2. 在导航窗格中，选择集群并滚动到网络设置 > 客户端 VPC 连接列表。
3. 选择要拒绝的连接，然后选择拒绝客户端 VPC 连接。
4. 确认要拒绝所选的客户端 VPC 连接。

要使用 API 拒绝托管式 VPC 连接，请使用 `RejectClientVpcConnection` API。

删除与 Amazon MSK 集群的托管式 VPC 连接

跨账户用户可以从客户端账户控制台中为 MSK 集群删除托管式 VPC 连接。集群所有者用户不拥有托管式 VPC 连接，因此无法从集群管理员账户中删除该连接。VPC 连接一经删除，就不会再产生费用。

使用控制台删除托管式 VPC 连接

1. 从客户端账户中，在 AWS Management Console 打开 Amazon MSK 控制台。
2. 在导航窗格中选择托管式 VPC 连接。
3. 从连接列表中选择要删除的连接。
4. 确认要删除 VPC 连接。

要使用 API 删除托管式 VPC 连接，请使用 `DeleteVpcConnection` API。

多 VPC 私有连接的权限

本节总结了使用多 VPC 私有连接功能的客户端和集群所需的权限。多 VPC 私有连接要求客户端管理员在将与 MSK 集群建立托管式 VPC 连接的每个客户端上创建权限。它还要求 MSK 集群管理员在 MSK 集群上启用 PrivateLink 连接，并选择身份验证方案来控制对集群的访问。

集群身份验证类型和主题访问权限

为针对您的 MSK 集群启用的身份验证方案开启多 VPC 私有连接功能。请参阅 多 VPC 私有连接的要求和限制。如果您将 MSK 集群配置为使用 SASL/SCRAM 身份验证方案，则必须提供 Apache Kafka ACL 属性 `allow.everyone.if.no.acl.found=false`。为集群设置 Apache Kafka ACL 后，请更新集群的配置，将该集群的属性 `allow.everyone.if.no.acl.found` 设置为 false。有关如何更新集群配置的信息，请参阅 Amazon MSK 配置操作。

跨账户集群策略权限

如果 Kafka 客户端所在的 AWS 账户与 MSK 集群不同，请将基于集群的策略附加到 MSK 集群，该策略授权客户端 root 用户进行跨账户连接。您可以使用 MSK 控制台中的 IAM policy 编辑器（集群安全设置 > 编辑集群策略）编辑多 VPC 集群策略，也可以使用以下 API 来管理集群策略：

PutClusterPolicy

将集群策略附加到集群。您可以使用此 API 来创建或更新指定的 MSK 集群策略。如果您要更新政策，则必须填写请求有效负载中的 currentVersion 字段。

GetClusterPolicy

检索附加到集群的集群策略文档的 JSON 文本。

DeleteClusterPolicy

删除集群策略。

以下是基本集群策略的 JSON 示例，类似于 MSK 控制台 IAM policy 编辑器中显示的策略。

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

与 MSK 集群的多 VPC 私有连接的客户端权限

要在 Kafka 客户端和 MSK 集群之间设置多 VPC 私有连接，客户端需要一个附加身份策略，以授予对客户端执行 `kafka:CreateVpcConnection`、`ec2:CreateTags` 和 `ec2:CreateVPCEndpoint` 操作的权限。以下是基本客户端身份策略的 JSON 示例，供您参考。

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

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

端口信息

使用以下端口号，以便 Amazon MSK 可以与客户端计算机通信：

- 要以明文与代理通信，请使用端口 9092。
- 要通过 TLS 加密与代理通信，请使用端口 9094 进行内部 AWS 访问，使用端口 9194 进行公共访问。
- 要通过 SASL/SCRAM 与经纪人通信，请使用端口 9096 进行内部访问，使用端口 9196 AWS 进行公共访问。
- 要与设置为使用的集群中的代理通信，请使用端口 9098 进行内部 AWS 访问the section called “IAM 访问控制”，使用端口 9198 进行公共访问。
- 要使用 TLS 加密与 Apache ZooKeeper 通信，请使用端口 2182。默认情况下，Apache ZooKeeper 节点使用端口 2181。

迁移到 Amazon MSK 集群

Amazon MSK 复制器可用于 MSK 集群迁移。请参阅什么是 Amazon MSK 复制器？。或者，您可以使用 Apache MirrorMaker 2.0 从非 MSK 集群迁移到 Amazon MSK 集群。有关如何执行此操作的示例，请参阅 Migrate an on-premises Apache Kafka cluster to Amazon MSK by using MirrorMaker。有关如何使用 MirrorMaker 的信息，请参阅 Apache Kafka 文档中的在集群间镜像数据。我们建议您在高可用性配置中设置 MirrorMaker。

使用 MirrorMaker 迁移到 MSK 集群时要执行的步骤概述

1. 创建目标 MSK 集群
2. 从目标集群所在的同一 Amazon VPC 中的 Amazon EC2 实例启动 MirrorMaker。
3. 检查 MirrorMaker 滞后。
4. 在 MirrorMaker 跟上之后，使用 MSK 集群引导代理将创建器和使用器重定向到新的集群。
5. 关闭 MirrorMaker。

将 Apache Kafka 集群迁移到 Amazon MSK

假定您有一个名为 CLUSTER_ONPREM 的 Apache Kafka 集群。该集群中已填充主题和数据。如果要将该集群迁移到新创建的名为 CLUSTER_AWSMSK 的 Amazon MSK 集群，此程序将提供您需要执行之步骤的高级视图。

将现有的 Apache Kafka 集群迁移到 Amazon MSK

1. 在 CLUSTER_AWSMSK 中，创建要迁移的所有主题。

   您无法在此步骤中使用 MirrorMaker，因为它不会自动使用正确的复制级别重新创建要迁移的主题。您可以使用与 CLUSTER_ONPREM 中相同的复制因子和分区数在 Amazon MSK 中创建主题。也可以创建具有不同的复制因子和分区数的主题。
2. 从具有 CLUSTER_ONPREM 的读访问权和 CLUSTER_AWSMSK 的写访问权的实例启动 MirrorMaker。
3. 运行以下命令以镜像所有主题：

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

在此命令中，config/mirrormaker-consumer.properties 指向 CLUSTER_ONPREM 中的引导代理；例如，bootstrap.servers=localhost:9092。config/mirrormaker-producer.properties 指向 CLUSTER_AWSMSK 中的引导代理；例如，bootstrap.servers=10.0.0.237:9092,10.0.2.196:9092,10.0.1.233:9092。

4. 使 MirrorMaker 在后台运行，并继续使用 CLUSTER_ONPREM。MirrorMaker 将镜像所有新数据。
5. 通过检查每个主题的最后一个偏移量与 MirrorMaker 正在使用的当前偏移量之间的滞后来检查镜像的进度。

   请记住，MirrorMaker 仅使用创建器和使用器。因此，您可以使用 kafka-consumer-groups.sh 工具检查滞后。要查找使用器组名称，请在 mirrormaker-consumer.properties 文件中查找 group.id，然后使用其值。如果文件中没有此类密钥，您可以创建它。例如，设置 group.id=mirrormaker-consumer-group。
6. 在 MirrorMaker 镜像完所有主题后，停止所有创建器和使用器，然后停止 MirrorMaker。然后，将创建器和使用器重定向到 CLUSTER_AWSMSK 集群，方式是更改该集群的创建器和使用器引导代理值。在 CLUSTER_AWSMSK 上重新启动所有创建器和使用器。

在 Amazon MSK 集群之间迁移

您可以使用 Apache MirrorMaker 2.0 从非 MSK 集群迁移到 MSK 集群。例如，您可以从一个版本的 Apache Kafka 迁移到另一个版本的 Apache Kafka。有关如何执行此操作的示例，请参阅 Migrate an on-premises Apache Kafka cluster to Amazon MSK by using MirrorMaker。Amazon MSK 复制器还可用于 MSK 集群迁移。有关 Amazon MSK 复制器的更多信息，请参阅 MSK 复制器。

MirrorMaker 1.0 最佳实践

此列表中的最佳实践适用于 MirrorMaker 1.0。

- 在目标集群上运行 MirrorMaker。这样一来，如果发生网络问题，消息仍在源集群中可用。如果您在源集群上运行 MirrorMaker，在创建器中缓冲事件且存在网络问题，ze 事件可能会丢失。
- 如果传输过程中需要加密，请在源集群中运行 MirrorMaker。
- 对于使用器，设置 auto.commit.enabled=false
- 对于创建器，设置
  - max.in.flight.requests.per.connection=1
  - retries=Int.Max_Value

  - acks=all
  - max.block.ms = Long.Max_Value
- 对于较高的创建器吞吐量：
  - 缓冲区消息和填充消息批处理 – 调整 buffer.memory、batch.size、linger.ms
  - 调整套接字缓冲区 – receive.buffer.bytes、send.buffer.bytes
- 为了避免数据丢失，请在源上关闭自动提交，以便 MirrorMaker 能够控制提交，这通常是在从目标集群收到 ack 后进行的。如果创建器的 acks=all 且目标集群的 min.insync.replicas 设置为大于 1，则在 MirrorMaker 使用器在源上提交偏移之前，这些消息将在目标上的多个代理上持久保存。
- 如果顺序很重要，则可将重试次数设置为 0。或者，对于生产环境，将最大传输中连接数设置为 1，以确保在批处理中途失败时，不会无序提交发出的批处理。这样一来，将重试发送的每个批处理，直到发出下一个批处理为止。如果 max.block.ms 未设置为最大值，并且如果创建器缓冲区已满，则可能会丢失数据（具体取决于其他一些设置）。这可以阻止和反压使用器。
- 对于高吞吐量
  - 增加 buffer.memory。
  - 增大批处理大小。
  - 调整 linger.ms 以允许填充批处理。这还可以实现更好的压缩、更少的网络带宽用量以及更少的集群存储。这会导致提高保留率。
  - 监控 CPU 和内存使用情况。
- 对于高使用器吞吐量
  - 增加每个 MirrorMaker 进程的线程/使用器数量 – num.streams。
  - 在增加线程数以实现高可用性之前，请先增加计算机之间的 MirrorMaker 进程数。
  - 依次增加同一台计算机和其他计算机（具有相同的组 ID）上的 MirrorMaker 进程数。
  - 隔离具有非常高的吞吐量的主题，并使用单独的 MirrorMaker 实例。
- 对于管理和配置
  - 使用 AWS CloudFormation 和配置管理工具，如 Chef 和 Ansible。
  - 使用 Amazon EFS 装载以确保可从所有 Amazon EC2 实例访问所有配置文件。
  - 使用容器来轻松扩展和管理 MirrorMaker 实例。
- 通常，要使 MirrorMaker中的创建器饱和，需要多个使用器。因此，请设置多个使用器。首先，在不同的计算机上设置使用器以实现高可用性。然后，扩展各个计算机以使每个分区有一个使用器，并且使用器在各个计算机之间均匀分配。
- 对于高吞吐量提取和传输，请调整接收和发送缓冲区，因为它们的默认值可能太小了。要获得最高性能，请确保流的总数 (num.streams) 与 MirrorMaker 正在尝试复制到目标集群的主题分区总数匹配。

MirrorMaker 2.* 的优势

- 可以利用 Apache Kafka Connect 框架和生态系统。
- 可以检测新主题和分区。
- 可以在集群之间自动同步主题配置。
- 支持“主动/主动”集群对以及任意数量的主动集群。
- 提供新指标，包括跨多个数据中心和集群的端到端复制延迟。
- 提供在集群之间迁移使用器所需的偏移量，并提供偏移量转换工具。
- 支持高级配置文件，以实现在一个位置指定多个集群和复制流，这与为每个 MirrorMaker 1.* 进程单独指定低级创建器/使用器属性不同。

监控 Amazon MSK 集群

Amazon MSK 可以通过多种方式帮助您监控 Amazon MSK 集群的状态。

- 当集群即将达到其存储容量限制时，Amazon MSK 自动向您发送存储容量警报，从而帮您监控磁盘存储容量。这些警报还就解决发现问题的最佳措施提供了建议。这有助于您在磁盘容量问题变得严重之前发现并快速解决它们。Amazon MSK 会自动将这些提醒发送到亚马逊 MSK 控制台 EventBridge、 AWS Health Dashboard亚马逊以及您 AWS 账户的电子邮件联系人。有关存储容量警报的更多信息，请参阅 Amazon MSK 存储容量警报。
- 亚马逊 MSK 收集 Apache Kafka 指标并将其发送到亚马逊，供您 CloudWatch 查看。有关 Apache Kafka 指标（包括 Amazon MSK 提供的指标）的更多信息，请参阅 Apache Kafka 文档中的监控。
- 您也可以使用开源监控应用程序 Prometheus 来监控 MSK 集群。有关 Prometheus 的信息，请参阅 Prometheus 文档中的概述。要了解如何使用 Prometheus 监控您的集群，请参阅the section called “Prometheus 的开源监控系统”。

主题

用于监控的 Amazon MSK 指标 CloudWatch

Amazon MSK 与亚马逊集成， CloudWatch 因此您可以收集、查看和分析亚马逊 MSK 集群的 CloudWatch指标。系统会自动收集您为 MSK 集群配置的指标并将其推送到。 CloudWatch您可以将 MSK 集群的监控级别设置为以下级别之一：DEFAULT、PER_BROKER、PER_TOPIC_PER_BROKER 或 PER_TOPIC_PER_PARTITION。以下部分中的表显示了在每个监控级别开始提供的所有指标。

> **Note**
>
> 在 3.6.0 及更高版本中，一些用于 CloudWatch 监控的 Amazon MSK 指标的名称已更改。请使用新名称来监控这些指标。对于名称已更改的指标，下表显示了 3.6.0 及更高版本中使用的名称，随后是 2.8.2.tiered 版本中使用的名称。

DEFAULT 级别指标免费。Amazon 定价页面中描述了其他指标的 CloudWatch定价。

DEFAULT 级别监控

下表中描述的指标在 DEFAULT 监控级别可用。这些指标是免费的。

DEFAULT 监控级别可用的指标

名称	可见时间	Dimensions	描述
ActiveControllerCount	在集群进入 ACTIVE 状态后。	集群名称	在任何给定时间，每个集群只能有一个控制器处于活动状态。
BurstBalance	在集群进入 ACTIVE 状态后。	集群名称、代理 ID	输入-输出突增积分剩余余额用于集群中的 EBS 卷。用它来调查延迟或吞吐量下降的情况。 当卷的基准性能超过最大突增性能时，BurstBalance 不会对 EBS 卷进行报告。有关更多信息，请参阅 I/O 积分和突增性能。
BytesInPerSec	在创建主题后。	集群名称、代理 ID、主题	每秒从客户端接收的字节数。此指标适用于每个代理和每个主题。
BytesOutPerSec	在创建主题后。	集群名称、代理 ID、主题	每秒发送到客户端的字节数。此指标适用于每个代理和每个主题。
ClientConnectionCount	在集群进入 ACTIVE 状态后。	集群名称、代理 ID、客	经过身份验证的活跃客户端连接数量。

名称	可见时间	Dimensions	描述
		户端身份验证	
ConnectionCount	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	经过身份验证、未经过身份验证以及代理间的活跃连接数量。
CPUCreditBalance	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	代理自启动后已累积获得的 CPU 积分数。在获得信用后，信用将在信用余额中累积；在花费信用后，将从信用余额中扣除信用。如果 CPU 积分余额用完，可能会对集群性能产生负面影响。您可以采取措施降低 CPU 负载。例如，您可以减少客户端请求的数量，或将代理类型更新为 M5 代理类型。
CpuIdle	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	CPU 空闲时间百分比。
CpuIoWait	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	待处理磁盘操作期间 CPU 空闲时间的百分比。
CpuSystem	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	内核空间中的 CPU 百分比。
CpuUser	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	用户空间中的 CPU 百分比。

名称	可见时间	Dimensions	描述
GlobalPartitionCount	在集群进入 ACTIVE 状态后。	集群名称	集群中所有主题的分区数量，不包括副本。由于GlobalPartitionCount 不包括副本，因此这些PartitionCount 值的总和可能高于 GlobalPartitionCount 主题的重复因子大于 1 的情况。
GlobalTopicCount	在集群进入 ACTIVE 状态后。	集群名称	集群中所有代理的主题总数。
EstimatedMaxTimeLag	在使用器组使用某个主题之后。	使用器组、主题	预计耗尽 MaxOffsetLag 的时间（以秒为单位）。
KafkaAppLogsDiskUsed	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	用于应用程序日志的磁盘空间的百分比。
KafkaDataLogsDiskUsed （Cluster Name, Broker ID 维度）	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	用于数据日志的磁盘空间的百分比。
KafkaDataLogsDiskUsed （Cluster Name 维度）	在集群进入 ACTIVE 状态后。	集群名称	用于数据日志的磁盘空间的百分比。
LeaderCount	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	每个代理的分区领导总数，不包括副本。

名称	可见时间	Dimensions	描述
MaxOffsetLag	在使用器组使用某个主题之后。	使用器组、主题	主题中所有分区之间的最大偏移延迟。
MemoryBuffered	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	代理的缓冲内存大小（以字节为单位）。
MemoryCached	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	代理的缓存内存大小（以字节为单位）。
MemoryFree	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	可供代理使用的可用内存大小（以字节为单位）。
HeapMemoryAfterGC	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	垃圾回收后使用的总堆内存百分比。
MemoryUsed	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	代理正在使用的内存大小（以字节为单位）。
MessagesInPerSec	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	代理每秒传入消息数。
NetworkRxDropped	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	丢弃的接收包的数量。
NetworkRxErrors	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	代理的网络接收错误数。

名称	可见时间	Dimensions	描述
NetworkRxPackets	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	代理收到的数据包的数量。
NetworkTxDropped	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	丢弃的传输包的数量。
NetworkTxErrors	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	代理的网络传输错误的数量。
NetworkTxPackets	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	代理传输的数据包的数量。
OfflinePartitionsCount	在集群进入 ACTIVE 状态后。	集群名称	集群中处于脱机状态的分区的总数。
PartitionCount	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	每个代理的主题分区总数，不包括副本。
ProduceTotalTimeMsMean	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	平均生成时间（以毫秒为单位）。
RequestBytesMean	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	代理的请求字节的平均数量。
RequestTime	在应用请求限制后。	集群名称，代理 ID	代理网络和 I/O 线程处理请求所花费的平均时间（以毫秒为单位）。

名称	可见时间	Dimensions	描述
RootDiskUsed	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	代理所使用的根磁盘的百分比。
SumOffsetLag	在使用器组使用某个主题之后。	使用器组、主题	主题中所有分区的聚合偏移延迟。
SwapFree	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	对代理可用的交换内存的大小（以字节为单位）。
SwapUsed	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	代理正在使用的交换内存的大小（以字节为单位）。
TrafficShaping	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	表示由于超出网络分配而形成（丢弃或排队）的数据包数量的高级指标。PER_BROKER 指标提供了更详细的信息。
UnderMinIsrPartitionCount	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	代理的未完全管理分区的数目。
UnderReplicatedPartitions	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	代理的未完全复制分区的数目。
ZooKeeperRequestLatencyMsMean	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	来自代理的 Apache ZooKeeper 请求的平均延迟（以毫秒为单位）。

名称	可见时间	Dimensions	描述
ZooKeeperSessionState	在集群进入 ACTIVE 状态后。	集群名称，代理 ID	经纪商 ZooKeeper 会话的连接状态可能是以下之一：NOT_CONNECTED：'0.0'，ASSOCIATING：'0.1'，正在连接：'0.5'，CONNECTEDREADONLY：'0.8'，已连接：'1.0'，已关闭：'5.0'，AUTH_FAILED：'10.0'。

PER_BROKER 级别监控

在将监控级别设置为 PER_BROKER 时，除了所有 DEFAULT 级别指标之外，您还将获得下表中描述的指标。您需要为下表中的指标付费，而 DEFAULT 级别指标仍免费。此表中的指标具有以下维度：集群名称、代理 ID。

在 **PER_BROKER** 监控级别开始提供的其他指标

名称	可见时间	描述
BwInAllowanceExceeded	在集群进入 ACTIVE 状态后。	因入站聚合带宽超过代理的最大值而形成的数据包的数量。
BwOutAllowanceExceeded	在集群进入 ACTIVE 状态后。	因出站聚合带宽超过代理的最大值而形成的数据包的数量。
ConnTrackAllowanceExceeded	在集群进入 ACTIVE 状态后。	因连接跟踪超过代理的最大值而形成的数据包的数量。连接跟踪与安全组相关，安全组会跟踪建立的每个连接，以确保返回数据包按预期交付。
ConnectionCloseRate	在集群进入 ACTIVE 状态后。	每个侦听器每秒关闭的连接数量。这个数字按每个侦听器聚合，并针对客户端侦听器进行筛选。

名称	可见时间	描述
ConnectionCreationRate	在集群进入 ACTIVE 状态后。	每个侦听器每秒建立的新连接数量。这个数字按每个侦听器聚合，并针对客户端侦听器进行筛选。
CpuCreditUsage	在集群进入 ACTIVE 状态后。	代理花掉的 CPU 积分数。如果 CPU 积分余额用完，可能会对集群性能产生负面影响。您可以采取措施降低 CPU 负载。例如，您可以减少客户端请求的数量，或将代理类型更新为 M5 代理类型。
FetchConsumerLocalTimeMsMean	在提供创建器/使用器后。	在领导处处理使用器请求所花费的平均时间（以毫秒为单位）。
FetchConsumerRequestQueueTimeMsMean	在提供创建器/使用器后。	使用器请求在请求队列中等待的平均时间（以毫秒为单位）。
FetchConsumerResponseQueueTimeMsMean	在提供创建器/使用器后。	使用器请求在响应队列中等待的平均时间（以毫秒为单位）。
FetchConsumerResponseSendTimeMsMean	在提供创建器/使用器后。	使用器发送响应所花费的平均时间（以毫秒为单位）。
FetchConsumerTotalTimeMsMean	在提供创建器/使用器后。	使用器从代理提取数据所花费的总平均时间（以毫秒为单位）。
FetchFollowerLocalTimeMsMean	在提供创建器/使用器后。	在领导处处理跟踪器请求所花费的平均时间（以毫秒为单位）。
FetchFollowerRequestQueueTimeMsMean	在提供创建器/使用器后。	跟踪器请求在请求队列中等待的平均时间（以毫秒为单位）。
FetchFollowerResponseQueueTimeMsMean	在提供创建器/使用器后。	跟踪器请求在响应队列中等待的平均时间（以毫秒为单位）。
FetchFollowerResponseSendTimeMsMean	在提供创建器/使用器后。	跟踪器发送响应所花费的平均时间（以毫秒为单位）。

名称	可见时间	描述
FetchFollowerTotalTimeMsMean	在提供创建器/使用器后。	跟踪器从代理提取数据所花费的总平均时间（以毫秒为单位）。
FetchMessageConversionsPerSec	在创建主题后。	代理每秒提取消息转换的次数。
FetchThrottleByteRate	在应用带宽限制后。	每秒的限制字节数。
FetchThrottleQueueSize	在应用带宽限制后。	限制队列中的消息数。
FetchThrottleTime	在应用带宽限制后。	平均提取限制时间（以毫秒为单位）。
IAMNumberOfConnectionRequests	在集群进入 ACTIVE 状态后。	每秒的 IAM 身份验证请求数。
IAMTooManyConnections	在集群进入 ACTIVE 状态后。	尝试的连接数超过 100。0 表示连接数在限制范围内。如果>0，则超过了油门限制，您需要减少连接数。
NetworkProcessorAvgIdlePercent	在集群进入 ACTIVE 状态后。	网络处理器处于空闲状态的时间的平均百分比。
PpsAllowanceExceeded	在集群进入 ACTIVE 状态后。	因双向 PPS 超过代理的最大值而形成的数据包的数量。
ProduceLocalTimeMsMean	在集群进入 ACTIVE 状态后。	在领导处处理请求所花费的平均时间（以毫秒为单位）。
ProduceMessageConversionsPerSec	在创建主题后。	代理每秒生成的消息转换数。
ProduceMessageConversionsTimeMsMean	在集群进入 ACTIVE 状态后。	消息格式转换所花费的平均时间（以毫秒为单位）。
ProduceRequestQueueTimeMsMean	在集群进入 ACTIVE 状态后。	请求消息在队列中所花费的平均时间（以毫秒为单位）。

名称	可见时间	描述
ProduceResponseQueueTimeMsMean	在集群进入 ACTIVE 状态后。	响应消息在队列中所花费的平均时间（以毫秒为单位）。
ProduceResponseSendTimeMsMean	在集群进入 ACTIVE 状态后。	发送响应消息所花费的平均时间（以毫秒为单位）。
ProduceThrottleByteRate	在应用带宽限制后。	每秒的限制字节数。
ProduceThrottleQueueSize	在应用带宽限制后。	限制队列中的消息数。
ProduceThrottleTime	在应用带宽限制后。	平均生成限制时间（以毫秒为单位）。
ProduceTotalTimeMsMean	在集群进入 ACTIVE 状态后。	平均生成时间（以毫秒为单位）。
RemoteFetchBytesPerSec (RemoteBytesInPerSec in v2.8.2.tiered)	在提供生成器/使用器后。	为响应使用器提取而从分层存储传输的总字节数。此指标包括影响下游数据传输流量的所有主题分区。类别：流量和错误率。这是一个 KIP-405 指标。
RemoteCopyBytesPerSec (RemoteBytesOutPerSec in v2.8.2.tiered)	在提供生成器/使用器后。	传输到分层存储的总字节数，包括来自日志段、索引和其他辅助文件的数据。此指标包括影响上游数据传输流量的所有主题分区。类别：流量和错误率。这是一个 KIP-405 指标。
RemoteLogManagerTasksAvgIdlePercent	在集群进入 ACTIVE 状态后。	远程日志管理器闲置时间的平均百分比。远程日志管理器将数据从代理传输到分层存储。类别：内部活动。这是一个 KIP-405 指标。

名称	可见时间	描述
RemoteLogReaderAvgIdlePercent	在集群进入 ACTIVE 状态后。	远程日志读取器闲置时间的平均百分比。远程日志读取器将数据从远程存储传输到代理，以响应使用器提取。类别：内部活动。这是一个 KIP-405 指标。
RemoteLogReaderTaskQueueSize	在集群进入 ACTIVE 状态后。	负责从分层存储中读取并等待安排的任务数量。类别：内部活动。这是一个 KIP-405 指标。
RemoteFetchErrorsPerSec (RemoteReadErrorPerSec in v2.8.2.tiered)	在集群进入 ACTIVE 状态后。	响应读取请求的总错误率，指定代理将这些请求发送到分层存储，以检索数据来响应使用器提取。此指标包括影响下游数据传输流量的所有主题分区。类别：流量和错误率。这是一个 KIP-405 指标。
RemoteFetchRequestsPerSec (RemoteReadRequestsPerSec in v2.8.2.tiered)	在集群进入 ACTIVE 状态后。	指定代理发送到分层存储以检索数据来响应使用器提取的读取请求的总数。此指标包括影响下游数据传输流量的所有主题分区。类别：流量和错误率。这是一个 KIP-405 指标。
RemoteCopyErrorsPerSec (RemoteWriteErrorPerSec in v2.8.2.tiered)	在集群进入 ACTIVE 状态后。	响应写入请求的总错误率，指定代理将这些请求发送到分层存储以向上游传输数据。此指标包括影响上游数据传输流量的所有主题分区。类别：流量和错误率。这是一个 KIP-405 指标。
ReplicationBytesInPerSec	在创建主题后。	每秒从其他代理接收的字节数。
ReplicationBytesOutPerSec	在创建主题后。	每秒发送到其他代理的字节数。

名称	可见时间	描述
RequestExemptFromThrottleTime	在应用请求限制后。	代理网络和 I/O 线程处理免受限制的请求所花费的平均时间（以毫秒为单位）。
RequestHandlerAvgIdlePercent	在集群进入 ACTIVE 状态后。	请求处理程序线程处于空闲状态的时间的平均百分比。
RequestThrottleQueueSize	在应用请求限制后。	限制队列中的消息数。
RequestThrottleTime	在应用请求限制后。	平均请求限制时间（以毫秒为单位）。
TcpConnections	在集群进入 ACTIVE 状态后。	显示设置了 SYN 标志的传入和传出 TCP 段的数量。
RemoteCopyLagBytes (TotalTierBytesLag in v2.8.2.tiered)	在创建主题后。	在代理上符合分层条件但尚未传输到分层存储的数据的总字节数。此指标显示了上游数据传输的效率。随着延迟增加，分层存储中无法持续存在的数据量也随之增加。类别：归档延迟。这不是一个 KIP-405 指标。
TrafficBytes	在集群进入 ACTIVE 状态后。	以总字节数显示客户端（生成器和使用器）与代理之间的网络流量。不报告代理之间的流量。
VolumeQueueLength	在集群进入 ACTIVE 状态后。	指定时间段内等待完成的读取和写入操作请求的数量。
VolumeReadBytes	在集群进入 ACTIVE 状态后。	在指定时间段内读取的字节数。
VolumeReadOps	在集群进入 ACTIVE 状态后。	在指定时间段内读取的操作数。
VolumeTotalReadTime	在集群进入 ACTIVE 状态后。	在指定时间段内完成所有读取操作耗费的总秒数。

名称	可见时间	描述
VolumeTotalWriteTime	在集群进入 ACTIVE 状态后。	在指定时间段内完成所有写入操作耗费的总秒数。
VolumeWriteBytes	在集群进入 ACTIVE 状态后。	在指定时间段内写入的字节数。
VolumeWriteOps	在集群进入 ACTIVE 状态后。	在指定时间段内写入操作的数量。

PER_TOPIC_PER_BROKER 级别监控

在将监控级别设置为 PER_TOPIC_PER_BROKER 时，除了 PER_BROKER 和 DEFAULT 级别的所有指标之外，您还将获得下表中描述的指标。仅 DEFAULT 级别指标是免费的。此表中的指标具有以下维度：集群名称、代理商 ID、主题。

> **Important**
>
> 对于使用 Apache Kafka 2.4.1 或更新版本的 Amazon MSK 集群，下表中的指标仅在其值首次变为非零后才会显示。例如，要查看 BytesInPerSec，一个或多个创建器必须先向集群发送数据。

在 **PER_TOPIC_PER_BROKER** 监控级别开始提供的其他指标

名称	可见时间	描述
FetchMessageConversionsPerSec	在创建主题后。	每秒转换的已提取消息的数量。
MessagesInPerSec	在创建主题后。	每秒接收的消息的数量。
ProduceMessageConversionsPerSec	在创建主题后。	已生成消息的每秒转换次数。
RemoteFetchBytesPerSec （RemoteBy	创建主题后，以及生成/使用主题时。	为响应使用器提取指定主题和代理而从分层存储传输的字节数。此指标包括影响指定代理上下游

名称	可见时间	描述
tesInPerSec in v2.8.2.tiered)		数据传输流量的所有主题分区。类别：流量和错误率。这是一个 KIP-405 指标。
RemoteCopyBytesPerSec (RemoteBytesOutPerSec in v2.8.2.tiered)	创建主题后，以及生成/使用主题时。	为指定主题和代理传输到分层存储的字节数。此指标包括影响指定代理上上游数据传输流量的所有主题分区。类别：流量和错误率。这是一个 KIP-405 指标。
RemoteFetchErrorsPerSec (RemoteReadErrorPerSec in v2.8.2.tiered)	创建主题后，以及生成/使用主题时。	响应读取请求的错误率，指定代理将这些请求发送到分层存储，以检索数据来响应使用器对指定主题的提取。此指标包括影响指定代理上下游数据传输流量的所有主题分区。类别：流量和错误率。这是一个 KIP-405 指标。
RemoteFetchRequestsPerSec (RemoteReadRequestsPerSec in v2.8.2.tiered)	创建主题后，以及生成/使用主题时。	指定代理发送到分层存储以检索数据来响应使用器对指定主题的提取的读取请求数。此指标包括影响指定代理上下游数据传输流量的所有主题分区。类别：流量和错误率。这是一个 KIP-405 指标。
RemoteCopyErrorsPerSec (RemoteWriteErrorPerSec in v2.8.2.tiered)	创建主题后，以及生成/使用主题时。	响应写入请求的错误率，指定代理将这些请求发送到分层存储以向上游传输数据。此指标包括影响指定代理上上游数据传输流量的所有主题分区。类别：流量和错误率。这是一个 KIP-405 指标。

PER_TOPIC_PER_PARTITION 级别监控

在将监控级别设置为 PER_TOPIC_PER_PARTITION 时，除了 PER_TOPIC_PER_BROKER、PER_BROKER 和 DEFAULT 级别的所有指标之外，您还将获得下表中所述的指标。仅 DEFAULT 级别指标是免费的。此表中的指标具有以下维度：使用器组、主题、分区。

在 **PER_TOPIC_PER_PARTITION** 监控级别开始提供的其他指标

名称	可见时间	描述
EstimatedTimeLag	在使用器组使用某个主题之后。	预计耗尽分区偏移延迟的时间（以秒为单位）。
OffsetLag	在使用器组使用某个主题之后。	分区级别使用器在偏移量方面的延迟。

使用查看亚马逊 MSK 指标 CloudWatch

您可以使用 CloudWatch 控制台、命令行或 CloudWatch API 监控 Amazon MSK 的指标。以下过程介绍如何使用这些不同的方式访问指标。

使用 CloudWatch 控制台访问指标

登录 AWS Management Console 并打开 CloudWatch 控制台，网址为 https://console.aws.amazon.com/cloudwatch/。

1. 在导航窗格中，选择指标。
2. 选择所有指标选项卡，然后选择 AWS/Kafka。
3. 要查看主题级别的指标，请选择 Topic, Broker ID, Cluster Name (主题、代理 ID、集群名称)；对于代理级别的指标，请选择 Broker ID, Cluster Name (代理 ID、集群名称)；对于集群级别的指标，请选择 Cluster Name (集群名称)。
4. （可选）在图表窗格中，选择统计数据和时间段，然后使用这些设置创建 CloudWatch 警报。

要访问指标，请使用 AWS CLI

使用列表指标和命令。get-metric-statistics

使用 CloudWatch CLI 访问指标

使用mon-list-metrics和mon-get-stats命令。

使用 CloudWatch API 访问指标

使用ListMetrics和GetMetricStatistics操作。

使用器延迟监控

通过监控使用器延迟，您可以识别速度缓慢或卡住的使用器，这些使用器没有跟上主题中可用的最新数据。必要时，您可以采取补救措施，例如扩展或重启这些使用器。要监控消费者延迟，您可以使用亚马逊 CloudWatch 或通过 Prometheus 开放监控。

使用器延迟指标可以量化写入主题的最新数据与应用程序读取的数据之间的差异。Amazon MSK 提供了以下消费者延迟指标，您可以通过亚马逊 CloudWatch 或通过 Prometheus 的开放监控获得这些指标：、、、和。`EstimatedMaxTimeLag` `EstimatedTimeLag` `MaxOffsetLag` `OffsetLag` `SumOffsetLag`有关这些指标的信息，请参阅 the section called “用于监控的 Amazon MSK 指标 CloudWatch”。

> **Note**
>
> 消费者延迟指标仅对处于稳定状态的消费者组可见。成功完成重新平衡后，消费组处于稳定状态，从而确保分区在使用者之间均匀分布。

Amazon MSK 支持采用 Apache Kafka 2.2.1 或更高版本的集群的使用器延迟指标。

Prometheus 的开源监控系统

您可以使用 Prometheus 监控 MSK 集群，Prometheus 是一种用于时间序列指标数据的开源监控系统。您可以使用 Prometheus 的远程写入功能，将这些数据发布到 Amazon Managed Service for Prometheus。您还可以使用与 Prometheus 格式的指标兼容的工具或与 Amazon MSK Open Monitoring 集成的工具，例如 Datadog、Lenses、New Relic 和 Sumo logic。开源监控系统可免费使用，但跨可用区传输数据需要付费。有关 Prometheus 的信息，请参阅 Prometheus 文档。

在启用开源监控系统的情况下创建 Amazon MSK 集群

使用 AWS Management Console

1. 登录并打开亚马逊 MSK 控制台，网址为 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。 AWS Management Console
2. 在 Monitoring (监控) 部分中，选中 Enable open monitoring with Prometheus (启用 Prometheus 开源监控系统) 旁边的复选框。
3. 在页面上的各部分中提供所需的信息，并查看所有可用的选项。
4. 选择创建集群。

使用 AWS CLI

- 调用 create-cluster 命令并指定其 open-monitoring 选项。启用 JmxExporter、NodeExporter 或两者。如果指定了 open-monitoring，则不能同时禁用这两个导出器。

使用 API

- 调用该CreateCluster操作并指定OpenMonitoring。启用 jmxExporter、nodeExporter 或两者。如果指定了 OpenMonitoring，则不能同时禁用这两个导出器。

为现有的 Amazon MSK 集群启用开源监控系统

要启用开源监控系统，请确保集群处于 ACTIVE 状态。

使用 AWS Management Console

1. 登录并打开亚马逊 MSK 控制台，网址为 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。 AWS Management Console
2. 选择要更新的集群的名称。这会将您引导至该集群的详细信息页面。
3. 在属性选项卡上，向下滚动以找到监控部分。
4. 选择编辑。
5. 选中 Enable open monitoring with Prometheus (启用 Prometheus 开源监控系统) 旁边的复选框。
6. 选择 保存更改。

使用 AWS CLI

- 调用 update-monitoring 命令并指定其 open-monitoring 选项。启用 JmxExporter、NodeExporter 或两者。如果指定了 open-monitoring，则不能同时禁用这两个导出器。

使用 API

- 调用该UpdateMonitoring操作并指定OpenMonitoring。启用 jmxExporter、nodeExporter 或两者。如果指定了 OpenMonitoring，则不能同时禁用这两个导出器。

在 Amazon EC2 实例上设置 Prometheus 主机

1. 从 https://prometheus.io/download/#prometheus 将 Prometheus 服务器下载到您的 Amazon EC2 实例上。
2. 将下载的文件解压缩到某个目录并转到该目录。
3. 使用以下内容创建名为 `prometheus.yml` 的文件。

```
# 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. 使用该ListNodes操作获取集群的代理列表。
5. 利用以下 JSON 创建名为 `targets.json` 的文件。将 *broker_dns_1*、*broker_dns_2* 和其余代理 DNS 名称替换为您在上一步中获取的代理 DNS 名称。包括您在上一步中获得的所有代理。Amazon MSK 对 JMX Exporter 使用端口 11001，对 Node Exporter 使用端口 11002。

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

6. 要在您的 Amazon EC2 实例上启动 Prometheus 服务器，请在您解压缩 Prometheus 文件并保存了 `prometheus.yml` 和 `targets.json` 的目录中运行以下命令。

```
./prometheus
```

7. 查找您在上一步中在其中运行 Prometheus 的 Amazon EC2 实例的 IPv4 公有 IP 地址。您在以下步骤中需要使用此公有 IP 地址。
8. 要访问 Prometheus Web UI，请打开可访问您的 Amazon EC2 实例的浏览器，然后转到 *Prometheus-Instance-Public-IP*`:9090`，其中的 *Prometheus-Instance-Public-IP* 是您在上一步中获得的公有 IP 地址。

Prometheus 指标

由 Apache Kafka 发送给 JMX 的所有指标都可通过 Prometheus 的开源监控系统访问。有关 Apache Kafka 指标的信息，请参阅 Apache Kafka 文档中的监控。除了 Apache Kafka 指标外，使用器延迟指标还可以在端口 11001 以 JMX MBean 名称 `kafka.consumer.group:type=ConsumerLagMetrics` 获得。您也可以使用 Prometheus Node Exporter 来获取代理端口 11002 的 CPU 和磁盘指标。

将 Prometheus 指标存储在 Amazon Managed Service for Prometheus 中

Amazon Managed Service for Prometheus 是一项与 Prometheus 兼容的监控和警报服务，可用于监控 Amazon MSK 集群。这是一项完全托管的服务，可自动扩缩指标的提取、存储、查询和警报。它还与 AWS 安全服务集成，使您可以快速、安全地访问数据。您可以使用开源 ProMQL 查询语言来查询指标并发出警报。

有关更多信息，请参阅开始使用 Amazon Managed Service Managed Service Prometheus。

Amazon MSK 存储容量警报

在 Amazon MSK 预配置集群上，可以选择集群的主存储容量。如果耗尽了预配置集群中代理的存储容量，可能会影响其生成和使用数据的能力，从而造成代价高昂的停机。Amazon MSK 提供的 CloudWatch指标可帮助您监控集群的存储容量。但是，为了便于您检测和解决存储容量问题，Amazon MSK 会自动向您发送动态集群存储容量警报。存储容量警报中包含有关采取短期和长期措施管理集群存储容量的建议。在 Amazon MSK 控制台中，可以使用警报中的快速链接立即采取建议的操作。

MSK 存储容量警报有两种类型：主动警报和补救警报。

- 主动（“需要操作”）存储容量警报会提醒您注意集群可能存在的存储问题。当 MSK 集群中的代理使用了 60% 或 80% 以上的磁盘存储容量时，您将收到有关受影响代理的主动警报。
- 当 MSK 集群中的一个代理磁盘存储容量用完时，补救（“需要采取关键操作”）存储容量警报要求您采取补救措施，修复严重的集群问题。

Amazon MSK 会自动将这些警报发送到亚马逊 MSK 控制台、Healt AWS h Das hboard EventBridge、Amazon 以及您 AWS 账户的电子邮件联系人。你也可以 EventBridge将亚马逊配置为向 Slack 或 New Relic 和 Datadog 等工具发送这些警报。

所有 MSK 预配置集群默认启用存储容量警报，而且无法关闭。所有提供 MSK 的区域均支持此功能。

监控 Amazon MSK 存储容量警报

您可以通过以下几种方式查看存储容量警报：

- 前往 Amazon MSK 控制台。存储容量警报将在集群警报窗格中显示 90 天。警报中包含解决磁盘存储容量问题的建议和单击链接操作。
- 使用ListClusters、ListClustersV2 或 DescribeClusterV2 API 查看集`CustomerActionStatus`群的所有警报。DescribeCluster

- 前往 AWS Health 控制面板查看来自 MSK 和其他 AWS 服务的警报。
- 设置 AWS Health API 和 Amazon，将警报通知路由 EventBridge到第三方平台，例如 Datadog 和 SI NewRelic ack。

将 Amazon MSK 与 LinkedIn 的适用于 Apache Kafka 的 Cruise Control 结合使用

您可以使用 LinkedIn 的 Cruise Control 来重新平衡 Amazon MSK 集群，检测和修复异常，并监控集群的状态和运行状况。

下载并构建 Cruise Control

1. 在与 Amazon MSK 集群相同的 Amazon VPC 中创建 Amazon EC2 实例。
2. 在上一步中创建的 Amazon EC2 实例上安装 Prometheus。记下私有 IP 和端口。默认端口号为 9090。有关如何配置 Prometheus 以聚合集群指标的信息，请参阅 the section called “Prometheus 的开源监控系统”。
3. 在 Amazon EC2 实例上下载 Cruise Control。（或者，如果您愿意，也可以针对 Cruise Control 使用单独的 Amazon EC2 实例。）对于具有 Apache Kafka 版本 2.4.* 的集群，请使用最新的 2.4.* Cruise Control 版本。如果集群的 Apache Kafka 版本早于 2.4.*，请使用最新的 2.0.* Cruise Control 版本。
4. 解压缩 Cruise Control 文件，然后转到解压缩后的文件夹。
5. 运行以下命令以安装 Git：

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

6. 运行以下命令以初始化本地存储库。将 *Your-Cruise-Control-Folder* 替换为当前文件夹（解压缩 Cruise Control 下载文件时获得的文件夹）的名称。

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

7. 运行以下命令以安装并构建源代码。

```
./gradlew jar copyDependantLibs
```

配置和运行 Cruise Control

1. 对 `config/cruisecontrol.properties` 文件进行以下更新。将示例引导服务器和 Apache ZooKeeeper 连接字符串替换为集群的值。要获取集群的这些字符串，您可以在控制台中查看集

群详细信息。或者，可以使用 GetBootstrapBrokers 和 DescribeCluster API 操作或其等效 CLI 操作。

```
# 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
zookeeper.connect=z-1.test-cluster.2skv42.c1.kafka.us-
east-1.amazonaws.com:2181,z-2.test-cluster.2skv42.c1.kafka.us-
east-1.amazonaws.com:2181,z-3.test-cluster.2skv42.c1.kafka.us-
east-1.amazonaws.com:2181

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

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

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

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

2. 编辑 `config/capacityCores.json` 文件以指定正确的磁盘大小、CPU 核心和网络输入/输出限制。您可以使用 DescribeCluster API 操作（或其等效 CLI 操作）来获取磁盘大小。有关 CPU 核心和网络输入/输出限制，请参阅 Amazon EC2 实例类型。

```
{
  "brokerCapacities": [
    {
      "brokerId": "-1",
      "capacity": {
        "DISK": "10000",
        "CPU": {
          "num.cores": "2"
        },
        "NW_IN": "5000000",
        "NW_OUT": "5000000"
      },
```

```
      "doc": "This is the default capacity. Capacity unit used for disk is in MB,
 cpu is in number of cores, network throughput is in KB."
    }
  ]
}
```

3. 您可以选择安装 Cruise Control UI。如需下载，请转到 Setting Up Cruise Control Frontend。
4. 运行以下命令以启动 Cruise Control。考虑使用类似 `screen` 或 `tmux` 的工具来保持长时间运行的会话处于开放状态。

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

5. 使用 Cruise Control API 或 UI 来确保 Cruise Control 拥有集群加载数据并提出重新平衡建议。获得有效的指标窗口可能需要几分钟时间。

Amazon MSK 限额

Amazon MSK 限额

- 每个账户最多可有 90 个代理，每个集群最多可有 30 个代理。要申请更高的配额，请前往 AWS 控制台 Support Center 并创建支持案例。
- 每个代理的最低存储空间为 1GiB。
- 每个代理的最高存储空间为 16384GiB。
- 在任何给定时间内，使用 the section called “IAM 访问控制” 的集群每个代理最多可以有 3000 个 TCP 连接。要提高此限制，您可以使用 Kafka AlterConfig API 或`kafka-configs.sh`工具调整或`listener.name.client_iam_public.max.connections`配置属性。`listener.name.client_iam.max.connections`值得注意的是，将任一属性增加到较高的值都可能导致不可用。
- 对 TCP 连接的限制。启用连接速率突发后，MSK 允许每秒 100 个连接。唯一的例外是 kafka.t3.small 实例类型，在启用连接速率突发的情况下，允许每秒 4 个连接。未启用连接速率突发的较旧集群将在群集修补时自动启用该功能。

  要处理连接失败时的重试，可以在客户端设置 `reconnect.backoff.ms` 配置参数。例如，如果您希望客户端在 1 秒钟后重试连接，请将 `reconnect.backoff.ms` 设置为 1000。有关更多信息，请参阅 Apache Kafka 文档中的 reconnect.backoff.ms。
- 每个账户具有最多 100 项配置。要请求限额调整，请前往 AWS 控制台支持中心并创建支持案例。
- 每个配置最多 50 个版本。
- 要更新配置或 MSK 集群的 Apache Kafka 版本，请首先确保每个代理的分区数低于 the section called “ 调整集群的大小：每个代理的分区数量” 中所述的限制。

MSK 复制器限额

- 每个账户最多 15 个 MSK 复制器。
- MSK Replicator 最多只能按排序顺序复制 750 个主题。如果您需要复制更多主题，我们建议您单独创建一个 Replicator。如果您需要为每个 Replicator 提供超过 750 个主题的支持，请前往 AWS 控制台 Support Center 并创建支持案例。您可以使用 “TopicCount” 指标监控正在复制的主题数量。
- 每个 MSK 复制器的最大入口吞吐量为每秒 1GB。要申请更高的配额，请前往 AWS 控制台 Support Center 并创建支持案例。

- MSK Replicator 记录大小-最大记录大小为 1MB（message.max.bytes）。要申请更高的配额，请前往 AWS 控制台 Support Center 并创建支持案例。

MSK Serverless 限额

> **Note**
>
> 如果您在配额限制方面遇到任何问题，请通过创建 AWS 支持案例与 S upport 联系。

除非另有说明，否则限制按每个集群计算。

维度	限额	限额违规结果
最大入口吞吐量	200MBps	减速，响应中提供节流持续时间
最大出口吞吐量	400MBps	减速，响应中提供节流持续时间
最长保留期	无限制	不适用
最大客户端连接数	3000	连接关闭
最大连接尝试次数	每秒 100 个	连接关闭
最大消息大小	8 MB	请求失败，并显示 ErrorCode：IN VALID_REQUEST
最大请求速率	每秒 15000 个	减速，响应中提供节流持续时间
主题管理 API 最大请求速率	每秒 2 个	减速，响应中提供节流持续时间
每次请求的最大获取字节数	55MB	请求失败，并显示 ErrorCode：IN VALID_REQUEST
最大使用器组数	500	JoinGroup 请求失败

维度	限额	限额违规结果
最大分区数（领导者）	非压缩主题为 2400。压缩主题为 120。要申请配额调整，请前往 AWS 控制台 Support Center 并创建支持案例。	请求失败，并显示 ErrorCode：IN VALID_REQUEST
分区创建和删除的最大速率	250（5 分钟）	请求失败，并显示 ErrorCode 为：吞吐量_配额_已超出
每个分区的最大入口吞吐量	5Mbps	减速，响应中提供节流持续时间
每个分区的最大出口吞吐量	10Mbps	减速，响应中提供节流持续时间
最大分区大小（压缩主题）	250GB	请求失败，并显示 ErrorCode 为：吞吐量_配额_已超出
每个无服务器集群的最大客户端 VPC 数量	5	
每个账户的最大无服务器集群数量	10. 要申请配额调整，请前往 AWS 控制台 Support Center 并创建支持案例。	
每个账户的最大 MSK 复制器数量	15 个复制器	

维度	限额	限额违规结果
每个 MSK 复制器的指标支持的最大主题数	MSK Replicator 最多只能按排序顺序复制 750 个主题。如果您需要复制更多主题，我们建议您单独创建一个 Replicator。如果您需要为每个 Replicator 提供超过 750 个主题的支持，请前往 AWS 控制台 Support Center 并创建支持案例。您可以使用 “TopicCount” 指标监控正在复制的主题数量。	
每个 MSK 复制器的最大入口吞吐量	每个 MSK 复制器每秒 1GB。要申请配额调整，请前往 AWS 控制台 Support Center 并创建支持案例。	
MSK Replicator 记录大小	最大记录大小为 1MB（消息.max.bytes）。要申请更高的配额，请前往 AWS 控制台 Support Center 并创建支持案例。	

MSK Connect 限额

- 最高 100 个自定义插件。
- 最高 100 个工作程序配置。
- 最多 60 个连接工作线程。如果将连接器设置为具有自动扩缩容量，则连接器所设置具有的工作程序最大数量就是 MSK Connect 用于计算账户限额的数量。
- 每个连接器最多 10 个工作程序。

要为 MSK Connect 申请更高的配额，请前往 AWS 控制台支持中心并创建支持案例。

Amazon MSK 资源

根据上下文可知，资源一词在 Amazon MSK 中有两种含义。在 API 上下文中，资源是一种可以在其上调用操作的结构。有关这些资源以及可在其上调用的操作的列表，请参阅《Amazon MSK API Reference》中的 Resources。在 the section called “IAM 访问控制” 上下文中，资源是可以允许或拒绝访问的实体，如 the section called “资源” 小节所定义。

MSK 集成

本节提供与 Amazon MSK 集成的 AWS 功能的参考。

主题

适用于 Amazon MSK 的 Amazon Athena 连接器

使用适用于 Amazon MSK 的 Amazon Athena 连接器，Amazon Athena 能够对 Apace Kafka 主题运行 SQL 查询。使用此连接器在 Athena 中以表的形式查看 Apache Kafka 主题，以行的形式查看消息。

有关更多信息，请参阅《Amazon Athena 用户指南》中的 Amazon Athena MSK 连接器。

Amazon Redshift 流数据摄取

Amazon Redshift 支持来自 Amazon MSK 的串流摄取。Amazon Redshift 串流摄取功能以低延迟、高速度的方式将流数据从 Amazon MSK 摄取到 Amazon Redshift 实体化视图中。由于不需要在 Amazon S3 中暂存数据，Amazon Redshift 能以更低的延迟和更低的存储成本摄取流数据。您可以使用 SQL 语句在 Amazon Redshift 集群上配置 Amazon Redshift 串流摄取，以对 Amazon MSK 主题进行身份验证和连接。

有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的串流摄取。

Firehose

亚马逊 MSK 与 Firehose 集成，提供了一种无服务器、无代码的解决方案，用于将流从 Apache Kafka 集群传输到亚马逊 S3 数据湖。Firehose 是一项流式提取、转换和加载 (ETL) 服务，它从你的 Amazon MSK Kafka 主题中读取数据，执行诸如转换到 Parquet 之类的转换，并将数据聚合并写入亚马逊 S3。只需在控制台中点击几下，您就可以设置 Firehose 直播，以便从 Kafka 主题中读取内容并传送到 S3 位置。无需编写代码，无需连接器应用程序，也无需预置资源。Firehose 会根据发布到 Kafka 主题的数据量自动进行扩展，您只需为从 Kafka 提取的字节付费。

有关此功能的更多信息，请参阅以下内容。

- 使用亚马逊 MSK 写入 Kinesis Data Firehose ——亚马逊 Kinesis Data Firehose 开发者指南中的亚马逊 Kinesis Data Firehose
- 博客：Amazon MSK Introduces Managed Data Delivery from Apache Kafka to Your Data Lake
- 实验：使用 Firehose 配送至亚马逊 S3

Apache Kafka 版本

创建 Amazon MSK 集群时，您可以指定您想要使用哪个 Apache Kafka 版本。您还可以更新现有集群的 Apache Kafka 版本。本章中的主题可帮助您了解 Kafka 版本支持的时间表和最佳实践建议。

主题

支持的 Apache Kafka 版本

Amazon Managed Streaming for Apache Kafka（Amazon MSK）支持以下 Apache Kafka 和 Amazon MSK 版本。Apache Kafka 社区在版本发布日期之后为其提供大约 12 个月的支持。有关更多详细信息，请查看 Apache Kafka EOL（生命周期终止）政策。

支持的 Kafka 版本

Kafka 版本	MSK 发布日期	支持终止日期
1.1.1	--	2024-06-05
2.1.0	--	2024-06-05
2.2.1	2019-07-31	2024-06-08
2.3.1	2019-12-19	2024-06-08
2.4.1	2020-04-02	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

Kafka 版本	MSK 发布日期	支持终止日期
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	--	2024-09-11
2.8.1	2022-10-28	2024-09-11
2.8.2 层	2022-10-28	--
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	--
3.5.1（推荐使用）	2023-09-26	--
3.6.0	2023-11-16	--

有关 Amazon MSK 版本支持政策的更多信息，请参阅亚马逊 MSK 版本支持政策。

Apache Kafka 版本 3.6.0（支持生产就绪的分层存储）

有关 Apache Kafka 版本 3.6.0（支持生产就绪的分层存储）的信息，请参阅 Apache Kafka 下载网站上的 Release Notes。

为了稳定起见，在本版本中，Amazon MSK 将继续使用和管理 Zookeeper 以进行仲裁管理。

Amazon MSK 分层存储版本 2.8.2.tiered

此版本是 Apache Kafka 版本 2.8.2 的仅限 Amazon MSK 版本，与开源 Apache Kafka 客户端兼容。

2.8.2.tiered 版本包含分层存储功能，该功能与 KIP-405 for Apache Kafka 中介绍的 API 兼容。有关 Amazon MSK 分层存储功能的更多信息，请参阅 分层存储。

Apache Kafka 版本 2.5.1

Apache Kafka 版本 2.5.1 包含多个错误修复和新功能，包括针对 Ap ZooKeeper ache 和管理客户端的传输加密。Amazon MSK 提供了 TLS ZooKeeper 终端节点，您可以通过DescribeCluster 操作进行查询。

该 DescribeCluster操作的输出包括ZookeeperConnectStringTls节点，其中列出了 TLS zookeeper 端点。

以下示例显示了 DescribeCluster 操作的响应 ZookeeperConnectStringTls 节点：

```
"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"
```

有关将 TLS 加密用于 Zookeeper 的信息，请参阅 在 Apache 中使用 TLS 安全性 ZooKeeper。

有关 Apache Kafka 版本 2.5.1 的更多信息，请参阅 Apache Kafka 下载网站上的 Release Notes。

Amazon MSK 错误修复版本 2.4.1.1

此版本是 Apache Kafka 版本 2.4.1 的仅限 Amazon MSK 的错误修复版本。此错误修复版本包含 KAFKA-9752 的修复程序，这是一个极少出现的问题，会导致使用器组不断重新平衡并保持 PreparingRebalance 状态。此问题会影响运行 Apache Kafka 版本 2.3.1 和 2.4.1 的集群。此版本包含社区制作的修复程序，可用于 Apache Kafka 版本 2.5.0。

> **Note**
>
> 运行版本 2.4.1.1 的 Amazon MSK 集群与兼容 Apache Kafka 版本 2.4.1 的任何 Apache Kafka 客户端兼容。

如果您更喜欢使用 Apache Kafka 2.4.1，建议您对新的 Amazon MSK 集群使用 MSK 错误修复版本 2.4.1.1。您可以将运行 Apache Kafka 版本 2.4.1 的现有集群更新为此版本，以加入此修复程序。有关升级现有集群的信息，请参阅 更新 Apache Kafka 版本。

要在不将集群升级到 2.4.1.1 版本的情况下解决此问题，请参阅 对 Amazon MSK 集群进行问题排查 指南的 使用器组卡滞在 PreparingRebalance 状态 部分。

Apache Kafka 版本 2.4.1（改用 2.4.1.1 版）

> **Note**
>
> 您无法再使用 Apache Kafka 版本 2.4.1 创建新的 MSK 集群。相反，您可以将 Amazon MSK 错误修复版本 2.4.1.1 与兼容 Apache Kafka 版本 2.4.1 的客户端结合使用。而且，如果已经拥有使用 Apache Kafka 版本 2.4.1 的 MSK 集群，建议您将其更新为使用 Apache Kafka 版本 2.4.1.1。

KIP-392 是 Apache Kafka 2.4.1 版中包含的重要 Kafka 改进建议之一。此项改进允许使用器从最近的副本提取。要使用此功能，请将使用器属性中的 `client.rack` 设置为使用器可用区的 ID。可用区 ID 的其中一个例子是 `use1-az1`。Amazon MSK 会将 `broker.rack` 设置为代理可用区 ID。您还必须将 `replica.selector.class` 配置属性设置为 `org.apache.kafka.common.replica.RackAwareReplicaSelector`，这是 Apache Kafka 提供的 rack 感知的一种实现方式。

当您使用此版本的 Apache Kafka 时，PER_TOPIC_PER_BROKER 监控级别中的指标仅在其值首次变为非零后才会显示。有关此问题的更多信息，请参阅 the section called “PER_TOPIC_PER_BROKER 级别监控”。

有关如何查找可用区 ID 的信息，请参阅 AWS Resource Access Manager 用户指南中的您的资源的可用区 ID。

有关设置配置属性的信息，请参阅配置。

有关 KIP-392 的更多信息，请参阅 Confluence 页面中的允许使用器从最近的副本提取。

有关 Apache Kafka 版本 2.4.1 的更多信息，请参阅 Apache Kafka 下载网站上的版本说明。

亚马逊 MSK 版本支持

本主题介绍的亚马逊 MSK 版本支持政策和过程更新 Apache Kafka 版本。如果您要升级 Kafka 版本，请遵循中版本升级的最佳实践概述的最佳实践。

亚马逊 MSK 版本支持政策

本节介绍亚马逊 MSK 支持的 Kafka 版本的支持政策。

- 所有 Kafka 版本均受支持，直至其终止支持日期。有关终止支持日期的详细信息，请参阅支持的 Apache Kafka 版本。在支持终止日期之前，将您的 MSK 集群升级到推荐的 Kafka 版本或更高版本。有关更新 Apache Kafka 版本的详细信息，请参阅。更新 Apache Kafka 版本在终止支持日期之后使用 Kafka 版本的集群会自动升级到推荐的 Kafka 版本。
- MSK 将逐步停止对使用 Kafka 版本且已公布支持终止日期的新创建集群的支持。

更新 Apache Kafka 版本

您可以将现有的 MSK 集群更新为较新版本的 Apache Kafka。您无法将它更新为较旧版本。在更新 MSK 集群的 Apache Kafka 版本时，还要检查您的客户端软件，以确保其版本允许您使用集群的新 Apache Kafka 版本的功能。Amazon MSK 仅更新服务器软件。它不会更新您的客户端。

有关如何在更新期间使集群高度可用的信息，请参阅the section called “构建高度可用的集群”。

> **Important**
>
> 对于超出 the section called “ 调整集群的大小：每个代理的分区数量” 中所述限制的 MSK 集群，您无法更新 Apache Kafka 版本。

使用更新 Apache Kafka 版本 AWS Management Console

1. 在 https://console.aws.amazon.com/msk/ 打开 Amazon MSK 控制台。
2. 选择要更新 Apache Kafka 版本的 MSK 集群。
3. 在属性选项卡上，在 Apache Kafka 版本部分中选择升级。

使用更新 Apache Kafka 版本 AWS CLI

1. 运行以下命令，*ClusterArn*替换为您在创建集群时获得的 Amazon 资源名称 (ARN)。如果您没有该集群的 ARN，可以通过列出所有集群来找到它。有关更多信息，请参阅 the section called “列出集群”。

```
aws kafka get-compatible-kafka-versions --cluster-arn ClusterArn
```

此命令的输出包括您可以将集群更新到的 Apache Kafka 版本的列表。其内容类似于以下示例。

```
{
    "CompatibleKafkaVersions": [
        {
            "SourceVersion": "2.2.1",
            "TargetVersions": [
                "2.3.1",
                "2.4.1",
                "2.4.1.1",
                "2.5.1"
            ]
        }
    ]
}
```

2. 运行以下命令，*ClusterArn*替换为您在创建集群时获得的 Amazon 资源名称 (ARN)。如果您没有该集群的 ARN，可以通过列出所有集群来找到它。有关更多信息，请参阅 the section called “列出集群”。

将 *Current-Cluster-Version* 替换为集群的当前版本。因为*TargetVersion*你可以从上一个命令的输出中指定任何目标版本。

> **Important**
>
> 集群版本不是简单的整数。要查找集群的当前版本，请使用DescribeCluster操作或 describe- AWS CLI cluster 命令。示例版本是 KTVPDKIKX0DER。

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

上一个命令的输出如以下 JSON 所示。

```
{

    "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. 要获得update-cluster-kafka-version操作结果，请运行以下命令，*ClusterOperationArn*替换为在命令输出中获得的 ARN。update-cluster-kafka-version

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

该 describe-cluster-operation 命令的输出如以下 JSON 示例所示。

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

如果 OperationState 的值为 UPDATE_IN_PROGRESS，请等待一段时间，然后再次运行 describe-cluster-operation 命令。操作完成后，OperationState 的值变为 UPDATE_COMPLETE。由于 Amazon MSK 完成操作所需的时间各不相同，您可能需要反复检查直到操作完成。

使用 API 更新 Apache Kafka 版本

1. 调用该GetCompatibleKafkaVersions操作以获取您可以将集群更新到的 Apache Kafka 版本列表。
2. 调用该UpdateClusterKafkaVersion操作将集群更新到兼容的 Apache Kafka 版本之一。

版本升级的最佳实践

为了确保在 Kafka 版本升级过程中执行的滚动更新期间的客户端连续性，请按以下方式查看客户端的配置和 Apache Kafka 主题：

- 2对于双可用区集群，将主题重复因子 (RF) 设置为最小值，将三可用区集群的最小值设置3为。RF 值为可能2导致在修补期间出现脱机分区。
- 将同步副本的最小值 (miniSR) 设置为最大值，RF - 1以确保分区副本集可以容忍一个副本离线或复制不足。

- 将客户端配置为使用多个代理连接字符串。如果开始修补支持客户端 I/O 的特定代理，则在客户端的连接字符串中包含多个代理可以进行故障转移。有关如何获取多个代理的连接字符串的信息，请参阅获取 Amazon MSK 集群的引导程序代理。
- 我们建议您将连接客户端升级到推荐版本或更高版本，以便从新版本中提供的功能中受益。客户端升级不受您的 MSK 集群 Kafka 版本的生命周期结束 (EOL) 日期的约束，也不需要在 EOL 日期之前完成。Apache Kafka 提供了双向客户端兼容性策略，允许较旧的客户端使用较新的集群，反之亦然。
- 使用 3.x.x 版本的 Kafka 客户端可能具有以下默认值：和。`acks=all` `enable.idempotence=true` `acks=all`与之前的默认值不同`acks=1`，它通过确保所有同步副本都确认生产请求来提供额外的耐久性。同样，的默认值`enable.idempotence`是之前的`false`。更改`enable.idempotence=true`为默认值可降低出现重复消息的可能性。这些更改被视为最佳实践设置，可能会在正常性能参数范围内引入少量额外延迟。
- 创建新的 MSK 集群时，请使用推荐的 Kafka 版本。使用推荐的 Kafka 版本可以让你受益于最新的 Kafka 和 MSK 功能。

对 Amazon MSK 集群进行问题排查

以下信息可帮助您排查 Amazon MSK 集群可能存在的问题。您也可以将问题发布到 AWS re:Post。

主题

由于复制过载，卷更换会导致磁盘饱和

在计划外的卷硬件故障期间，Amazon MSK 可能会用新实例替换该卷。Kafka 通过复制集群中其他代理的分区来重新填充新卷。一旦分区被复制并赶上，它们就有资格获得领导和同步副本 (ISR) 成员资格。

问题

在从卷更换中恢复过来的代理中，一些大小不一的分区可能会先于其他分区重新联机。这可能会出现问题，因为这些分区可能提供来自同一个代理的流量，而这些代理仍在赶上（复制）其他分区。这种复制流量有时会使底层卷吞吐量限制饱和，默认情况下为每秒 250 MiB。当这种饱和度发生时，任何已经被捕获的分区都将受到影响，从而导致任何与被捕分区共享 ISR 的代理在集群中出现延迟（而不仅仅是由于远程 ack 而导致的领导分区`acks=all`）。此问题在较大的集群中更为常见，这些群集的分区数量较多，大小各不相同。

建议

- 要改善复制 I/O 状态，请确保最佳实践线程设置到位。
- 要降低底层容量饱和的可能性，请启用具有更高吞吐量的预配置存储。对于高吞吐量复制案例，建议将 500 MiB/s 的最小吞吐量值设置为 500 MiB/s，但实际所需的值会因吞吐量和用例而异。 预置存储吞吐量。
- 要最大限度地减少复制压力，`num.replica.fetchers`请降低到默认值2。

使用器组卡滞在 `PreparingRebalance` 状态

如果一个或多个使用器组卡滞在一个永久的再平衡状态，则原因可能是 Apache Kafka 问题 KAFKA-9752，这会影响 Apache Kafka 版本 2.3.1 和 2.4.1。

要解决此问题，建议您将集群升级到 Amazon MSK 错误修复版本 2.4.1.1，其中包含针对此问题的修复程序。有关将现有集群更新到 Amazon MSK 错误修复版本 2.4.1.1 的信息，请参阅 更新 Apache Kafka 版本。

在不将集群升级到 Amazon MSK 错误修复版本 2.4.1.1 的情况下解决此问题的方法是，设置要使用 静态成员协议 的 Kafka 客户端，或者 识别并重启 卡住的使用器组的协调代理节点。

实现静态成员协议

要在客户端中实现静态成员协议，请执行以下操作：

1. 将 Kafka 使用器配置的 `group.instance.id` 属性设置为可识别组中使用器的静态字符串。
2. 确保配置的其他实例已更新为使用静态字符串。
3. 将更改部署到您的 Kafka 使用器。

如果将客户端配置中的会话超时设置为允许使用器在不过早触发使用器组重新平衡的情况下恢复的持续时间，则使用静态成员协议会更有效。例如，如果您的使用器应用程序可以容忍 5 分钟不可用，则会话超时的合理值为 4 分钟，而不是默认的 10 秒。

> **Note**
>
> 使用静态成员协议只会降低遇到此问题的可能性。即使使用静态成员协议，您仍可能遇到此问题。

重启协调代理节点

要重启协调代理节点，请执行以下操作：

1. 使用 `kafka-consumer-groups.sh` 命令识别组协调器。
2. 使用 RebootBrokerAPI 操作重新启动卡住的消费者组的群组协调器。

向 Amazon CloudWatch 日志传送代理日志时出错

当您尝试将集群设置为向 Amazon Logs 发送代理 CloudWatch 日志时，可能会遇到两个例外情况之一。

如果遇到 `InvalidInput.LengthOfCloudWatchResourcePolicyLimitExceeded` 异常，请重试，但使用以 `/aws/vendedlogs/` 开头的日志组。有关更多信息，请参阅启用从某些 Amazon Web Services 进行日志记录。

如果您遇到异`InvalidInput.NumberOfCloudWatchResourcePoliciesLimitExceeded`常，请选择您账户中的现有 Ama CloudWatch zon Logs 政策，并在其中附加以下 JSON。

```
{"Sid":"AWSLogDeliveryWrite","Effect":"Allow","Principal":
{"Service":"delivery.logs.amazonaws.com"},"Action":
["logs:CreateLogStream","logs:PutLogEvents"],"Resource":["*"]}
```

如果您尝试将上述 JSON 附加到现有策略中，但收到错误消息，提示您已达到所选策略的最大长度，请尝试将 JSON 附加到您的另一个 Amazon CloudWatch Logs 策略中。将 JSON 附加到现有策略后，请再次尝试将代理日志传输设置为 Amazon Logs。 CloudWatch

无默认安全组

如果您尝试创建集群，并收到错误指示没有默认安全组，则可能是因为您使用的是共享 VPC。请向管理员申请向您授予描述此 VPC 上的安全组的权限，然后重试。有关允许此操作的策略示例，请参阅 Amazon EC2：允许以编程方式在控制台中管理与特定 VPC 关联的 EC2 安全组。

集群显示卡在 CREATING 状态

有时，集群创建可能需要长达 30 分钟。请等待 30 分钟，然后再次检查集群的状态。

集群状态从 CREATING 变为 FAILED

请尝试再次创建集群。

集群状态为 ACTIVE，但生成器无法发送数据，或者使用器无法接收数据

- 如果集群创建成功（集群状态为 ACTIVE），但您无法发送或接收数据，请确保生成器和使用器应用程序有权访问集群。有关更多信息，请参阅the section called “步骤 3：创建客户端计算机”中的指南。

- 如果您的生产器和使用器有权访问集群，但仍出现生成和使用数据问题，原因可能是 KAFKA-7697，这会影响 Apache Kafka 2.1.0 版本，并可能导致一个或多个代理发生死锁。请考虑迁移到 Apache Kafka 2.2.1，该版本不受此错误影响。有关如何迁移的信息，请参阅迁移。

AWS CLI 无法识别 Amazon MSK

如果您已 AWS CLI 安装但它无法识别 Amazon MSK 命令，请 AWS CLI 将您的命令升级到最新版本。有关如何升级的详细说明 AWS CLI，请参阅安装 AWS Command Line Interface。有关如何使用运行 Amazon MSK 命令的信息，请参阅工作方式。 AWS CLI

分区脱机或副本不同步

这些可能是磁盘空间不足的症状。请参阅 the section called “磁盘空间不足”。

磁盘空间不足

请参阅以下有关管理磁盘空间的最佳实践：the section called “监控磁盘空间”和the section called “调整数据保留参数”。

内存不足

如果您发现 MemoryUsed 指标太高或 MemoryFree 太低，这并不意味着存在问题。Apache Kafka 的设计初衷是充分利用内存，并以最佳方式管理内存。

制片人获得 NotLeaderForPartitionException

这往往是临时错误。将生成器的 retries 配置参数设置为高于其当前值的值。

复制中的分区（URP）大于零

UnderReplicatedPartitions 指标是要监控的重要指标。在正常运行的 MSK 集群中，此指标的值为 0。如果它大于零，这可能是由以下某个原因所致。

- 如果 UnderReplicatedPartitions 是峰值，问题可能在于该集群的大小配置不合适，无法处理传入和传出流量。请参阅 最佳实践。
- 如果 UnderReplicatedPartitions 始终大于 0（包括在低流量期间），问题可能在于您设置了限制性 ACL，该 ACL 未向代理授予主题访问权限。要复制分区，必须向代理授予 READ 和 DESCRIBE 主题的权限。默认情况下，将随 READ 授权一起授予 DESCRIBE 权限。有关设置 ACL 的信息，请参阅 Apache Kafka 文档中的授权和 ACL 。

集群中有名为 __amazon_msk_canary 和 __amazon_msk_canary_state 的主题

您可能会看到，MSK 集群有一个名为 __amazon_msk_canary 的主题，而另一个主题的名称为 __amazon_msk_canary_state。这些是 Amazon MSK 创建并用于集群运行状况和诊断指标的内部主题。这些主题无法删除，不过大小可以忽略不计。

分区复制失败

确保您尚未在 CLUSTER_ACTIONS 上设置 ACL。

无法访问已开启公共访问权限的集群

如果您的集群已开启公共访问权限，但您仍然无法通过互联网访问它，请按照以下步骤操作：

1. 确保集群安全组的入站规则允许您的 IP 地址和集群端口。有关集群端口号的列表，请参阅 the section called “端口信息”。还要确保安全组的出站规则允许出站通信。有关安全组及其入站和出站规则的更多信息，请参阅《Amazon VPC 用户指南》中的您的 VPC 的安全组。
2. 确保集群 VPC 网络 ACL 的入站规则中允许您的 IP 地址和集群端口。与安全组不同，网络 ACL 无状态。这意味着您必须配置入站和出站规则。在出站规则中，允许所有流量（端口范围：0-65535）发送到您的 IP 地址。有关更多信息，请参阅《Amazon VPC 用户指南》中的添加和删除规则。
3. 确保您使用的是公共访问引导代理字符串来访问集群。开启了公共访问权限的 MSK 集群有两个不同的引导代理字符串，一个用于公共访问，另一个用于从 AWS内部访问。有关更多信息，请参阅 the section called “使用获取引导程序代理 AWS Management Console”。

无法从内部访问集群 AWS：网络问题

如果您的 Apache Kafka 应用程序无法与 MSK 集群成功通信，可以先执行以下连接测试。

1. 使用the section called “获取引导代理”中介绍的方法之一获取引导代理的地址。
2. 在以下命令中，将 *bootstrap-broker* 替换为您在上一步中获得的某个代理地址。如果将集群设置为使用 TLS 身份验证，则将 *port-number* 替换为 9094。如果集群不使用 TLS 身份验证，请将 *port-number* 替换为 9092。从客户端计算机运行命令。

```
telnet bootstrap-broker port-number
```

3. 对所有引导代理重复运行上面的命令。
4. 使用中the section called “获取 Apache ZooKeeper 连接字符串”描述的任何方法获取集群 Apache ZooKeeper 节点的地址。
5. 在客户端计算机上运行以下命令，将 *Apache ZooKeeper-nod* e 替换为在上一步中获得的其中一个 Apache ZooKeeper 节点的地址。数字 2181 是端口号。对所有 Apache ZooKeeper 节点重复此操作。

```
telnet Apache-ZooKeeper-node 2181
```

如果客户端计算机能够访问代理和 Apache ZooKeeper 节点，则表示没有连接问题。在这种情况下，可以运行以下命令来检查 Apache Kafka 客户端是否设置正确。要获取 *bootstrap-brokers*，可使用the section called “获取引导代理”中介绍的方法之一。将 *topic* 替换为您的主题名称。

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

如果上一个命令成功，则表示客户端设置正确。如果仍然无法从应用程序创建和使用，请在应用程序级别调试问题。

如果客户端计算机无法访问代理和 Apache ZooKeeper 节点，请参阅以下小节以获取基于您的客户端计算机设置的指导。

同一 VPC 中的 Amazon EC2 客户端和 MSK 集群

如果客户端计算机与 MSK 集群位于同一 VPC 中，请确保集群安全组具有接受来自客户端计算机安全组的流量的入站规则。有关设置这些规则的信息，请参阅安全组规则。有关如何从与集群位于同一 VPC 中的 Amazon EC2 实例访问集群的示例，请参阅 开始使用。

位于不同 VPC 中的 Amazon EC2 客户端和 MSK 集群

如果客户端计算机和集群位于两个不同的 VPC 中，请确保满足以下条件：

- 这两个 VPC 是对等连接的。
- 对等连接处于活动状态。
- 这两个 VPC 的路由表已正确设置。

有关 VPC 对等连接的信息，请参阅使用 VPC 对等连接。

本地客户端

如果本地客户端设置为使用连接到 MSK 集群 AWS VPN，请确保满足以下条件：

- VPN 连接状态为 UP。有关如何检查 VPN 连接状态的信息，请参阅如何检查 VPN 隧道的当前状态？。
- 集群 VPC 的路由表包含目标格式为 `Virtual private gateway(vgw-xxxxxxxx)` 的本地 CIDR 的路由。
- MSK 集群的安全组允许端口 2181、端口 9092（如果您的集群接受明文流量）和端口 9094（如果您的集群接受 TLS 加密的流量）上的流量传输。

有关更多 AWS VPN 故障排除指南，请参阅 Client VPN 故障排除。

AWS Direct Connect

如果客户端使用 AWS Direct Connect，请参阅故障排除 AWS Direct Connect。

如果上述问题排查指导未能解决此问题，请确保没有防火墙阻止网络流量。若要进一步调试，请使用 `tcpdump` 和 `Wireshark` 等工具来分析流量，并确保流量到达 MSK 集群。

身份验证失败：连接次数过多

`Failed authentication ... Too many connects` 错误表明代理正在保护自己，因为一个或多个 IAM 客户端正试图以激进的速度连接到它。为帮助代理接受更高的新 IAM 连接速率，您可以增加 `reconnect.backoff.ms` 配置参数。

要详细了解每个代理的新连接的速率限制，请参阅 Amazon MSK 限额 页面。

MSK Serverless：集群创建失败

如果您尝试创建 MSK Serverless 集群，但工作流程失败，则您可能无权创建 VPC 端点。通过允许 `ec2:CreateVpcEndpoint` 操作，验证您的管理员是否已授予您创建 VPC 端点的权限。

有关执行所有 Amazon MSK 操作所需的完整权限列表，请参阅 AWS 托管策略：AmazonMSKFullAccess。

最佳实践

本主题概述使用 Amazon MSK 时应遵循的一些最佳实践。

调整集群的大小：每个代理的分区数量

下表显示了建议的每个代理的分区数量（包括领导副本和跟随者副本）。

代理类型	建议的每个代理的分区数量（包括领导副本和跟随者副本）。
kafka.t3.small	300
kafka.m5.large 或 kafka.m5.xlarge	1000
kafka.m5.2xlarge	2000
kafka.m5.4xlarge 、kafka.m5.8xlarge 、kafka.m5.12xlarge 、kafka.m5.16xlarge 或 kafka.m5.24xlarge	4000
kafka.m7g.large 或 kafka.m7g.xlarge	1000
kafka.m7g.2xlarge	2000
kafka.m7g.4xlarge 、kafka.m7g.8xlarge kafka.m7g.12xlarge 、或 kafka.m7g.16xlarge	4000

如果每个代理的分区数量超过建议值，并且您的集群过载，则可能会阻止您执行以下操作：

- 更新集群配置
- 更新集群的 Apache Kafka 版本
- 将集群更新为较小的代理类型

- 将 AWS Secrets Manager 密钥与具有 SASL/SCRAM 身份验证的集群相关联

大量分区还可能导致 Prometheus 抓取 CloudWatch 和抓取时缺少 Kafka 指标。

有关选择分区数的指导，请参阅 Apache Kafka 支持每个集群 20 万个分区。我们还建议您执行自己的测试，以确定适合您代理的类型。有关不同代理类型的更多信息，请参阅the section called “代理类型”。

调整集群的大小：每个集群的代理数量

要确定 MSK 集群的适当代理数量并了解成本，请参阅 MSK Sizing and Pricing 电子表格。此电子表格提供了与类似的、自我管理的基于 EC2 的 Apache Kafka 集群相比，估计的 MSK 集群大小和相关 Amazon MSK 成本。有关电子表格中的输入参数的更多信息，请将鼠标指针悬停在参数描述的上方。此表提供的是保守估计值，为新集群提供了一个起点。集群的性能、大小和成本取决于您的用例，建议您通过实际测试进行验证。

要了解底层基础架构如何影响 Apache Kafka 性能，请参阅大数据博客中的调整您的 Apache Kafka 集群规模以优化性能和成本的最佳实践。 AWS 这篇博客文章提供了有关如何调整集群大小以满足吞吐量、可用性和延迟要求的信息。它还提供了诸如何时应纵向扩展，何时应横向扩展等问题的答案，以及有关如何持续验证生产集群大小的指导。

优化 m5.4xl、m7g.4xl 或更大实例的集群吞吐量

使用 m5.4xl、m7g.4xl 或更大的实例时，您可以通过调整 num.io.threads 和 num.network.threads 配置来优化集群吞吐量。

Num.io.threads 是代理用于处理请求的线程数。添加更多线程（不超过实例类型支持的 CPU 核心数量）有助于提高集群的吞吐量。

Num.network.threads 是代理用于接收所有传入请求和返回响应的线程数。网络线程将传入请求放在请求队列中，以供 io.threads 处理。将 num.network.threads 设置为实例类型支持的 CPU 核心数量的一半，即可充分使用新的实例类型。

> **Important**
>
> 如果不先增加 num.io.threads，请勿增加 num.network.threads，因为这可能会导致与队列饱和相关的拥塞。

推荐设置

实例类型	num.io.threads 的推荐值	num.network.threads 的推荐值
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

使用最新的 Kafka AdminClient 来避免主题 ID 不匹配问题

当您使用低于 2.8.0 的 Kafka 版本并带有标志`--zookeeper`为使用 Kafka AdminClient 版本 2.8.0 或更高版本的集群增加或重新分配主题分区时，主题的 ID 会丢失（错误：与分区的主题 ID 不匹配）。请注意，`--zookeeper` 标志在 Kafka 2.5 中已弃用，并从 Kafka 3.0 开始删除。请参阅 Upgrading to 2.5.0 from any version 0.8.x through 2.4.x。

为防止主题 ID 不匹配，请使用 Kafka 客户端版本 2.8.0 或更高版本进行 Kafka 管理员操作。或者，2.5 及更高版本的客户端可以使用 `--bootstrap-servers` 标志代替 `--zookeeper` 标志。

构建高度可用的集群

使用以下建议，以便在更新期间（例如更新代理类型或 Apache Kafka 版本时）或 Amazon MSK 更换代理时，保持 MSK 集群的高可用性。

- 设置三可用区集群。
- 确保复制因子（RF）至少为 3。请注意，在滚动更新期间，RF 为 1 可能会导致分区离线；而 RF 为 2 可能会导致数据丢失。

- 将最小同步副本数 (minISR) 设置为最多 RF - 1。minISR 等于 RF 可能会阻止在滚动更新期间生成到集群。当一个副本处于脱机状态时，minISR 为 2 使三向复制主题可用。
- 确保客户端连接字符串至少包含来自每个可用区的一个代理。在客户端的连接字符串中具有多个代理，则可在特定代理脱机进行更新时实现失效转移。有关如何获取具有多个代理的连接字符串的信息，请参阅the section called “获取引导代理”。

监控 CPU 使用率

Amazon MSK 强烈建议您将代理的总 CPU 使用率（定义为 `CPU User + CPU System`）保持在 60% 以下。当集群的总 CPU 可用率至少达到 40% 时，Apache Kafka 可以在必要时在集群中的代理之间重新分配 CPU 负载。例如，当 Amazon MSK 检测到代理故障并从中恢复时，就有必要这样做；在这种情况下，Amazon MSK 会执行自动维护，如进行修补。另一个例子是当用户请求更改代理类型或升级版本时；在这两种情况下，Amazon MSK 会部署滚动工作流程，一次让一个代理离线。当具有领导分区的代理离线时，Apache Kafka 会重新分配分区领导权，以将工作重新分配给集群中的其他代理。通过遵循此最佳实践，您可以确保集群中有足够的 CPU 余量来容忍此类操作事件。

您可以使用 Amazon CloudWatch 指标数学来创建复合指标，即`CPU User + CPU System`。设置当复合指标达到 60% 的平均 CPU 利用率时触发的警报。触发此警报时，请使用以下选项之一扩展集群：

- 选项 1（推荐）：将您的代理类型更新为下一个较大的类型。例如，如果当前类型为 `kafka.m5.large`，则更新集群以使用 `kafka.m5.xlarge`。请记住，当您更新集群中的代理类型时，Amazon MSK 会以滚动方式使代理离线，并暂时将分区领导权重新分配给其他代理。每个代理的规模更新通常需要 10-15 分钟。
- 选项 2：如果主题中的所有消息都是从使用轮询写入的生成器那里摄取的（换句话说，消息没有密钥，顺序对使用器来说并不重要），请通过添加代理来扩展集群。还要向吞吐量最高的现有主题添加分区。接下来，使用 `kafka-topics.sh --describe` 来确保将新添加的分区分配给新代理。与前一个选项相比，此选项的主要优点是您可以更精细地管理资源和成本。此外，如果 CPU 负载明显超过 60%，则可使用此选项，因为这种形式的扩展通常不会导致现有代理的负载增加。
- 选项 3：通过添加代理来扩展集群，然后使用名为 `kafka-reassign-partitions.sh` 的分区重新分配工具来重新分配现有分区。但是，如果您使用此选项，则在重新分配分区后，集群将需要花费资源将数据从一个代理复制到另一个代理。与前两个选项相比，这可能会在一开始显著增加集群的负载。因此，Amazon MSK 不建议在 CPU 利用率高于 70% 时使用此选项，因为复制会导致额外的 CPU 负载和网络流量。仅当前两个选项不可行时，Amazon MSK 才建议使用此选项。

其他建议：

- 作为负载分配的代理，监控每个代理的 CPU 总利用率。如果代理的 CPU 利用率一直不均衡，则可能表明集群内的负载分布不均。Amazon MSK 建议使用 Cruise Control 通过分区分配持续管理负载分配。
- 监控生成和使用延迟。生成和使用延迟会随着 CPU 利用率呈线性增加。
- JMX 抓取间隔：如果您使用 Prometheus 功能启用开源监控系统，则建议您为 Prometheus 主机配置 (prometheus.yml) 使用 60 秒或更长的抓取间隔 (scrape_interval: 60s)。降低抓取间隔可能会导致集群上的 CPU 使用率过高。

监控磁盘空间

为避免存储消息的磁盘空间不足，请创建`KafkaDataLogsDiskUsed`监控指标的 CloudWatch 警报。当此指标的值达到或超过 85% 时，请执行下列一项或多项操作：

- 使用 the section called “自动扩缩”。您也可以手动增加代理存储空间，如 the section called “手动扩展” 中所述。
- 缩短消息保留期或减小日志大小。有关如何做到这一点的信息，请参阅the section called “调整数据保留参数”。
- 删除未使用的主题。

有关如何设置和使用警报的信息，请参阅使用 Amazon CloudWatch 警报。有关 Amazon MSK 指标的完整列表，请参阅 监控集群。

调整数据保留参数

使用消息不会将其从日志中删除。要定期释放磁盘空间，您可以明确指定一个保留时间段，即消息在日志中保留的时间。您也可以指定保留日志大小。当达到保留时间段或保留日志大小时，Apache Kafka 会开始从日志中删除非活动段。

要在集群级别指定保留策略，请设置以下一个或多个参数：`log.retention.hours`、`log.retention.minutes`、`log.retention.ms` 或 `log.retention.bytes`。有关更多信息，请参阅 the section called “自定义 配置”。

您也可以在主题级别指定保留参数：

- 要为每个主题指定一个保留时间段，请使用以下命令。

```
kafka-configs.sh --bootstrap-server $bs --alter --entity-type topics --entity-
name TopicName --add-config retention.ms=DesiredRetentionTimePeriod
```

- 要为每个主题指定一个保留日志大小，请使用以下命令。

```
kafka-configs.sh --bootstrap-server $bs --alter --entity-type topics --entity-
name TopicName --add-config retention.bytes=DesiredRetentionLogSize
```

您在主题级别指定的保留参数优先于集群级别参数。

在不正常关闭后加快日志恢复

在不正常关闭后，代理可能需要一段时间才能重新启动，因为它需进行日志恢复。默认情况下，Kafka 仅对每个日志目录使用一个线程来执行此恢复。例如，如果您有成千上万个分区，则日志恢复可能需要数个小时才能完成。为加快日志恢复，建议使用配置属性 `num.recovery.threads.per.data.dir` 增加线程数量。您可以将它设置为 CPU 核心的数量。

监控 Apache Kafka 内存

建议您监控 Apache Kafka 使用的内存。否则，集群可能会变得不可用。

要确定 Apache Kafka 使用了多少内存，您可以监控 `HeapMemoryAfterGC` 指标。`HeapMemoryAfterGC` 是垃圾回收后使用的总堆内存百分比。我们建议您创建一个 CloudWatch 警报，当`HeapMemoryAfterGC`增幅超过 60% 时会采取行动。

可用于减少内存使用的步骤会有所不同，具体取决于您配置 Apache Kafka 的方式。例如，如果您使用事务性消息传递，则可以将 Apache Kafka 配置中的 `transactional.id.expiration.ms` 值从 `604800000` 毫秒减少到 `86400000` 毫秒（从 7 天减少到 1 天）。这减少了每个事务的内存占用。

请勿添加非 MSK 代理

如果您使用 Apache ZooKeeper 命令添加代理，则这些代理不会被添加到您的 MSK 集群中，并且您的 Apache ZooKeeper 将包含有关该集群的错误信息。这可能会导致丢失数据。有关受支持的集群操作，请参阅工作方式。

启用传输中加密

有关传输中加密以及如何启用此加密的信息，请参阅the section called “传输中加密”。

重新分配分区

要将分区移动到同一集群上的不同代理，您可以使用名为 `kafka-reassign-partitions.sh` 的分区重新分配工具。例如，在添加新代理来扩展集群后，您可以通过将分区重新分配给新代理来重新使集群达到平衡。有关如何向集群添加代理的信息，请参阅the section called “扩展集群”。有关分区重新分配工具的信息，请参阅 Apache Kafka 文档中的扩展集群。

MSK 复制器的 API 操作

Amazon Managed Streaming for Apache Kafka REST API 包括以下针对 MSK 复制器的操作。

- CreateReplicator

  创建复制器。
- DeleteReplicator

  删除复制器。
- DescribeReplicator

  描述复制器。
- ListReplicators

  列出复制器。
- UpdateReplicationInfo

  更新复制器的复制信息。

MSK 复制器的 API 资源

主题

V1 Replicators

URI

`/replication/v1/replicators`

HTTP 方法

GET

操作 ID：`ListReplicators`

列出复制器。

查询参数

名称	Type	必需	描述
`replicatorNameFilter`	String	False	返回以给定名称开头的 MSK 复制器。
`nextToken`	String	False	分页结果标记。当操作结果被截断时，调用将在响应中返回 `NextToken` 。要获取下一批结果，请在下次请求中提供此令牌。
`maxResults`	String	False	要在响应中返回的最大结果数（默认情

名称	Type	必需	描述
			况下，每次 API 调用最多返回 100 个结果）。如果结果更多，则响应中包含 NextToken 参数。

响应

状态代码	响应模型	描述
200	ListReplicatorsResponse	HTTP 状态代码 200：OK。
400	None	请求无效，因为输入错误。请更正输入，然后重新提交。
401	None	请求未经授权。无法验证提供的凭证。
403	None	禁止访问。请检查凭证，然后重试请求。
404	None	由于输入错误，找不到资源。请更正输入，然后重试请求。
429	None	429 响应
500	None	出现意外内部服务器错误。重试请求可能会解决该问题。
503	None	503 响应

POST

操作 ID：CreateReplicator

创建复制器。

响应

状态代码	响应模型	描述
200	CreateReplicatorResponse	HTTP 状态代码 200：OK。
400	None	请求无效，因为输入错误。请更正输入，然后重新提交。
401	None	请求未经授权。无法验证提供的凭证。
403	None	禁止访问。请检查凭证，然后重试请求。
404	None	由于输入错误，找不到资源。请更正输入，然后重试请求。
409	None	此集群名称已存在。请使用其他名称重试请求。
429	None	429 响应
500	None	出现意外内部服务器错误。重试请求可能会解决该问题。
503	None	503 响应

OPTIONS

通过返回正确标头来启用 CORS

响应

状态代码	响应模型	描述
200	None	CORS 方法的默认响应

架构

请求正文

POST 架构

```
{
  "replicatorName": "string",
  "serviceExecutionRoleArn": "string",
  "replicationInfoList": [
    {
      "consumerGroupReplication": {
        "consumerGroupsToExclude": [
          "string"
        ],
        "detectAndCopyNewConsumerGroups": boolean,
        "consumerGroupsToReplicate": [
          "string"
        ],
        "synchroniseConsumerGroupOffsets": boolean
      },
      "targetCompressionType": enum,
      "topicReplication": {
        "copyAccessControlListsForTopics": boolean,
        "detectAndCopyNewTopics": boolean,
        "copyTopicConfigurations": boolean,
        "topicsToReplicate": [
          "string"
        ],
        "topicsToExclude": [
          "string"
        ]
      },
      "sourceKafkaClusterArn": "string",
      "targetKafkaClusterArn": "string"
    }
  ],
  "description": "string",
  "kafkaClusters": [
    {
      "amazonMskCluster": {
        "mskClusterArn": "string"
      },
```

```
      "vpcConfig": {
        "securityGroupIds": [
          "string"
        ],
        "subnetIds": [
          "string"
        ]
      }
    }
  ],
  "tags": {
  }
}
```

响应正文

ListReplicatorsResponse 架构

```
{
  "nextToken": "string",
  "replicators": [
    {
      "replicatorArn": "string",
      "replicatorName": "string",
      "creationTime": "string",
      "kafkaClustersSummary": [
        {
          "kafkaClusterAlias": "string",
          "amazonMskCluster": {
            "mskClusterArn": "string"
          }
        }
      ],
      "replicatorState": enum,
      "isReplicatorReference": boolean,
      "replicationInfoSummaryList": [
        {
          "sourceKafkaClusterAlias": "string",
          "targetKafkaClusterAlias": "string"
        }
      ],
      "replicatorResourceArn": "string",
      "currentVersion": "string"
```

```
    }
  ]
}
```

CreateReplicatorResponse 架构

```
{
  "replicatorArn": "string",
  "replicatorName": "string",
  "replicatorState": enum
}
```

属性

AmazonMskCluster

Amazon MSK 集群的详细信息。

mskClusterArn

Amazon MSK 集群的 Amazon 资源名称（ARN）。

类型：字符串
必需：True

ConsumerGroupReplication

有关使用器组复制的详细信息。

consumerGroupsToExclude

指定不应复制的使用器组的正则表达式模式列表。

类型：string 类型的数组
必填项：False
MaxLength：256

detectAndCopyNewConsumerGroups

启用使用器组与 MSK 复制器目标集群的同步。

类型：布尔值
必填项：False

consumerGroupsToReplicate

表示要复制的使用器组的正则表达式模式列表。

类型：string 类型的数组
必需：True
MaxLength：256

synchroniseConsumerGroupOffsets

启用使用器组偏移与 MSK 复制器目标集群的同步。转换后的偏移将写入主题 __consumer_offsets。

类型：布尔值
必填项：False

CreateReplicatorRequest

复制器的请求正文。

replicatorName

复制器的名称。允许使用字母数字字符和“-”。

类型：字符串
必需：True
模式：^[0-9A-Za-z][0-9A-Za-z-]{0,}$
MinLength: 1
MaxLength：128

serviceExecutionRoleArn

复制器用于访问客户账户中资源（例如源集群和目标集群）的 IAM 角色的 Amazon 资源名称（ARN）

类型：字符串
必需：True

replicationInfoList

复制配置列表，其中每个配置都以给定源集群到目标集群复制流程为目标。

类型：ReplicationInfo 类型的数组
必需：True

description

复制器的摘要描述。

类型：字符串
必填项：False
MaxLength：1024

kafkaClusters

用于设置复制的源/目标的 Kafka 集群。

类型：KafkaCluster 类型的数组
必需：True

tags

要附加到已创建复制器的标签列表。

类型：对象
必填项：False

CreateReplicatorResponse

返回有关创建的 MSK 复制器的信息。

replicatorArn

MSK 复制器的 Amazon 资源名称（ARN）。

类型：字符串
必填项：False

replicatorName

客户提供的 MSK 复制器的名称。

类型：字符串
必填项：False

replicatorState

MSK 复制器的状态。

类型:ReplicatorState
必填项：False

KafkaCluster

有关用作复制的源/目标的 Kafka 集群的信息。

amazonMskCluster

Amazon MSK 集群的详细信息。

类型:AmazonMskCluster
必需：True

vpcConfig

与 Apache Kafka 集群有网络连接的 Amazon VPC 的详细信息。

类型:KafkaClusterClientVpcConfig
必需：True

KafkaClusterClientVpcConfig

与 Kafka 集群有网络连接的 Amazon VPC 的详细信息。

securityGroupIds

要附加到代理节点的 ENI 的安全组。

类型：string 类型的数组
必填项：False

subnetIds

客户端 VPC 中可连接到的子网列表。

类型：string 类型的数组
必需：True

KafkaClusterSummary

有关用作复制的源/目标的 Kafka 集群的摘要信息。

kafkaClusterAlias

Kafka 集群的别名。用于为已复制主题的名称添加前缀。

类型：字符串
必填项：False

amazonMskCluster

Amazon MSK 集群的详细信息。

类型:AmazonMskCluster
必需：False

ListReplicatorsResponse

响应包含一个包含 MSK Replicator 信息的数组，以及响应 NextToken 是否被截断。

nextToken

如果的响应被截断， ListReplicators 则它会在响应 NextToken 中返回 a。这 NextToken 应在随后的请求中发送给 ListReplicators。

类型：字符串
必填项：False

replicators

包含账户中每个 MSK 复制器的信息的列表。

类型：ReplicatorSummary 类型的数组
必填项：False

ReplicationInfo

指定在 MSK 复制器源和目标 Kafka 集群之间复制的配置。

consumerGroupReplication

与使用器组复制相关的配置。

类型:ConsumerGroupReplication
必需：True

targetCompressionType

向 MSK 复制器目标集群生成记录时要使用的压缩类型。

类型:TargetCompressionType
必需：True

topicReplication

与主题复制相关的配置。

类型:TopicReplication
必需：True

sourceKafkaClusterArn

MSK 复制器源 Kafka 集群的 Amazon 资源名称（ARN）。

类型：字符串
必需：True

targetKafkaClusterArn

MSK 复制器目标 Kafka 集群的 Amazon 资源名称（ARN）。

类型：字符串
必需：True

ReplicationInfoSummary

集群间复制的摘要信息。

sourceKafkaClusterAlias

MSK 复制器源 Kafka 集群的别名。

类型：字符串
必填项：False

targetKafkaClusterAlias

MSK 复制器目标 Kafka 集群的别名。

类型：字符串
必填项：False

ReplicatorState

MSK 复制器的状态。

```
RUNNING
CREATING
UPDATING
DELETING
FAILED
```

ReplicatorSummary

有关 MSK 复制器的信息。

replicatorArn

MSK 复制器的 Amazon 资源名称（ARN）。

类型：字符串
必填项：False

replicatorName

MSK 复制器的名称。

类型：字符串
必填项：False

creationTime

创建 MSK 复制器的时间。

类型：字符串
必填项：False

kafkaClustersSummary

用于设置复制的源/目标的 Kafka 集群。

类型：KafkaClusterSummary 类型的数组
必填项：False

replicatorState

MSK 复制器的状态。

类型:ReplicatorState
必填项：False

isReplicatorReference

表明此资源是否是 MSK 复制器引用。

类型：布尔值
必填项：False

replicationInfoSummaryList

集群间复制的摘要信息列表。

类型：ReplicationInfoSummary 类型的数组
必填项：False

replicatorResourceArn

创建复制器的区域中 MSK 复制器资源的 Amazon 资源名称（ARN）。

类型：字符串
必填项：False

currentVersion

MSK 复制器当前版本。

类型：字符串
必填项：False

TargetCompressionType

向目标集群生成记录时要使用的压缩类型。

```
NONE
GZIP
SNAPPY
LZ4
ZSTD
```

TopicReplication

有关主题复制的详细信息。

copyAccessControlListsForTopics

是否定期配置远程主题 ACL 以匹配其对应的上游主题。

类型：布尔值
必填项：False

detectAndCopyNewTopics

是否定期检查新主题和分区。

类型：布尔值
必填项：False

copyTopicConfigurations

是否定期配置远程主题以匹配其对应的上游主题。

类型：布尔值
必填项：False

topicsToReplicate

指定要复制的主题的正则表达式模式列表。

类型：string 类型的数组
必需：True
MaxLength: 249

topicsToExclude

指定不应复制的主题的正则表达式模式列表。

类型：string 类型的数组
必填项：False
MaxLength: 249

另请参阅

有关在特定语言的 AWS SDK 和参考中使用此 API 的更多信息，请参阅以下内容：

ListReplicators

- AWS 命令行界面
- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS 适用于 JavaScript
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包
- AWS 适用于 Ruby V3 的 SDK

CreateReplicator

- AWS 命令行界面
- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS 适用于 JavaScript
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包
- AWS 适用于 Ruby V3 的 SDK

V1 Replicators replicatorArn

URI

`/replication/v1/replicators/`*`replicatorArn`*

HTTP 方法

GET

操作 ID：`DescribeReplicator`

描述复制器。

路径参数

名称	Type	必需	描述
`ReplicatorArn`	String	True	要描述的 MSK 复制器的Amazon 资源名称（ARN）。

响应

状态代码	响应模型	描述
200	DescribeReplicatorResponse	HTTP 状态代码 200：OK。
400	None	请求无效，因为输入错误。请更正输入，然后重新提交。
401	None	请求未经授权。无法验证提供的凭证。
403	None	禁止访问。请检查凭证，然后重试请求。
404	None	由于输入错误，找不到资源。请更正输入，然后重试请求。
429	None	429 响应
500	None	出现意外内部服务器错误。重试请求可能会解决该问题。
503	None	503 响应

DELETE

操作 ID：`DeleteReplicator`

删除复制器。

路径参数

名称	Type	必需	描述
`ReplicatorArn`	String	True	要描述的 MSK 复制器的Amazon 资源名称（ARN）。

查询参数

名称	Type	必需	描述
`currentVersion`	String	False	MSK 集群的当前版本。

响应

状态代码	响应模型	描述
200	DeleteReplicatorResponse	HTTP 状态代码 200：OK。
400	None	请求无效，因为输入错误。请更正输入，然后重新提交。
401	None	请求未经授权。无法验证提供的凭证。
403	None	禁止访问。请检查凭证，然后重试请求。
404	None	由于输入错误，找不到资源。请更正输入，然后重试请求。

状态代码	响应模型	描述
429	None	429 响应
500	None	出现意外内部服务器错误。重试请求可能会解决该问题。
503	None	503 响应

OPTIONS

通过返回正确标头来启用 CORS

路径参数

名称	Type	必需	描述
ReplicatorArn	String	True	要描述的 MSK 复制器的Amazon 资源名称（ARN）。

响应

状态代码	响应模型	描述
200	None	CORS 方法的默认响应

架构

响应正文

DescribeReplicatorResponse 架构

```
{
  "replicatorArn": "string",
  "creationTime": "string",
  "kafkaClusters": [
    {
      "kafkaClusterAlias": "string",
```

```
      "amazonMskCluster": {
        "mskClusterArn": "string"
      },
      "vpcConfig": {
        "securityGroupIds": [
          "string"
        ],
        "subnetIds": [
          "string"
        ]
      }
    }
  ],
  "currentVersion": "string",
  "tags": {
  },
  "replicatorDescription": "string",
  "replicatorName": "string",
  "serviceExecutionRoleArn": "string",
  "replicationInfoList": [
    {
      "consumerGroupReplication": {
        "consumerGroupsToExclude": [
          "string"
        ],
        "detectAndCopyNewConsumerGroups": boolean,
        "consumerGroupsToReplicate": [
          "string"
        ],
        "synchroniseConsumerGroupOffsets": boolean
      },
      "targetCompressionType": enum,
      "sourceKafkaClusterAlias": "string",
      "topicReplication": {
        "copyAccessControlListsForTopics": boolean,
        "detectAndCopyNewTopics": boolean,
        "copyTopicConfigurations": boolean,
        "topicsToReplicate": [
          "string"
        ],
        "topicsToExclude": [
          "string"
        ]
      },
```

```
      "targetKafkaClusterAlias": "string"
    }
  ],
  "stateInfo": {
    "code": "string",
    "message": "string"
  },
  "replicatorState": enum,
  "isReplicatorReference": boolean,
  "replicatorResourceArn": "string"
}
```

DeleteReplicatorResponse 架构

```
{
  "replicatorArn": "string",
  "replicatorState": enum
}
```

属性

AmazonMskCluster

Amazon MSK 集群的详细信息。

mskClusterArn

Amazon MSK 集群的 Amazon 资源名称（ARN）。

类型：字符串
必需：True

ConsumerGroupReplication

有关使用器组复制的详细信息。

consumerGroupsToExclude

指定不应复制的使用器组的正则表达式模式列表。

类型：string 类型的数组

必填项：False
MaxLength：256

detectAndCopyNewConsumerGroups

启用使用器组与 MSK 复制器目标集群的同步。

类型：布尔值
必填项：False

consumerGroupsToReplicate

表示要复制的使用器组的正则表达式模式列表。

类型：string 类型的数组
必需：True
MaxLength：256

synchroniseConsumerGroupOffsets

启用使用器组偏移与 MSK 复制器目标集群的同步。转换后的偏移将写入主题 __consumer_offsets。

类型：布尔值
必填项：False

DeleteReplicatorResponse

返回有关删除的 MSK 复制器的信息。

replicatorArn

MSK 复制器的 Amazon 资源名称（ARN）。

类型：字符串
必填项：False

replicatorState

MSK 复制器的状态。

类型:ReplicatorState
必需：False

DescribeReplicatorResponse

的响应正文 DescribeReplicator。

replicatorArn

MSK 复制器的 Amazon 资源名称（ARN）。

类型：字符串
必填项：False

creationTime

创建 MSK 复制器的时间。

类型：字符串
必填项：False

kafkaClusters

用于设置复制的源/目标的 Kafka 集群。

类型：KafkaClusterDescription 类型的数组
必填项：False

currentVersion

MSK 复制器的当前版本号。

类型：字符串
必填项：False

tags

附加到 MSK 复制器的标签列表。

类型：对象
必填项：False

replicatorDescription

MSK 复制器的描述。

类型：字符串
必填项：False

replicatorName

MSK 复制器的名称。

类型：字符串
必填项：False

serviceExecutionRoleArn

MSK 复制器用于访问客户账户中资源（例如源集群和目标集群）的 IAM 角色的 Amazon 资源名称（ARN）。

类型：字符串
必填项：False

replicationInfoList

复制配置列表，其中每个配置都以给定源集群到目标集群复制流程为目标。

类型：ReplicationInfoDescription 类型的数组
必填项：False

stateInfo

有关 MSK 复制器状态的详细信息。

类型:ReplicationStateInfo
必填项：False

replicatorState

MSK 复制器的状态。

类型:ReplicatorState
必填项：False

isReplicatorReference

表明此资源是否是 MSK 复制器引用。

类型：布尔值
必填项：False

replicatorResourceArn

创建复制器的区域中 MSK 复制器资源的 Amazon 资源名称（ARN）。

类型：字符串
必填项：False

KafkaClusterClientVpcConfig

与 Kafka 集群有网络连接的 Amazon VPC 的详细信息。

securityGroupIds

要附加到代理节点的 ENI 的安全组。

类型：string 类型的数组
必填项：False

subnetIds

客户端 VPC 中可连接到的子网列表。

类型：string 类型的数组
必需：True

KafkaClusterDescription

有关用作复制的源/目标的 Kafka 集群的信息。

kafkaClusterAlias

Kafka 集群的别名。用于为已复制主题的名称添加前缀。

类型：字符串
必填项：False

amazonMskCluster

Amazon MSK 集群的详细信息。

类型:AmazonMskCluster
必填项：False

vpcConfig

与 Kafka 集群有网络连接的 Amazon VPC 的详细信息。

类型:KafkaClusterClientVpcConfig
必需：False

ReplicationInfoDescription

指定在 MSK Replicator 源和目标 Kafka 集群之间进行复制的配置（sourceKafkaCluster别名-> targetKafkaCluster 别名）。

consumerGroupReplication

与使用器组复制相关的配置。

类型:ConsumerGroupReplication
必填项：False

targetCompressionType

向 MSK 复制器目标集群生成记录时要使用的压缩类型。

类型:TargetCompressionType
必填项：False

sourceKafkaClusterAlias

MSK 复制器源 Kafka 集群的别名。

类型：字符串
必填项：False

topicReplication

与主题复制相关的配置。

类型:TopicReplication
必填项：False

targetKafkaClusterAlias

MSK 复制器目标 Kafka 集群的别名。

类型：字符串
必填项：False

ReplicationStateInfo

有关 MSK 复制器状态的详细信息。

code

描述 MSK 复制器当前状态的代码。

类型：字符串
必填项：False

message

描述 MSK 复制器状态的消息。

类型：字符串

必填项：False

ReplicatorState

MSK 复制器的状态。

RUNNING
CREATING
UPDATING
DELETING
FAILED

TargetCompressionType

向目标集群生成记录时要使用的压缩类型。

NONE
GZIP
SNAPPY
LZ4
ZSTD

TopicReplication

有关主题复制的详细信息。

copyAccessControlListsForTopics

是否定期配置远程主题 ACL 以匹配其对应的上游主题。

类型：布尔值
必填项：False

detectAndCopyNewTopics

是否定期检查新主题和分区。

类型：布尔值
必填项：False

copyTopicConfigurations

是否定期配置远程主题以匹配其对应的上游主题。

类型：布尔值
必填项：False

topicsToReplicate

指定要复制的主题的正则表达式模式列表。

类型：string 类型的数组
必需：True
MaxLength: 249

topicsToExclude

指定不应复制的主题的正则表达式模式列表。

类型：string 类型的数组
必填项：False
MaxLength: 249

另请参阅

有关在特定语言的 AWS SDK 和参考中使用此 API 的更多信息，请参阅以下内容：

DescribeReplicator

- AWS 命令行界面
- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS 适用于 JavaScript
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包

- AWS 适用于 Ruby V3 的 SDK

DeleteReplicator

- AWS 命令行界面
- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS 适用于 JavaScript
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包
- AWS 适用于 Ruby V3 的 SDK

V1 Replicators replicatorArn Replication-info

URI

`/replication/v1/replicators/replicatorArn/replication-info`

HTTP 方法

PUT

操作 ID：`UpdateReplicationInfo`

更新复制器的复制信息。

路径参数

名称	Type	必需	描述
ReplicatorArn	String	True	要描述的 MSK 复制器的Amazon 资源名称（ARN）。

响应

状态代码	响应模型	描述
200	UpdateReplicationInfoResponse	HTTP 状态代码 200：OK。
400	None	请求无效，因为输入错误。请更正输入，然后重新提交。
401	None	请求未经授权。无法验证提供的凭证。
403	None	禁止访问。请检查凭证，然后重试请求。
404	None	由于输入错误，找不到资源。请更正输入，然后重试请求。
429	None	429 响应
500	None	出现意外内部服务器错误。重试请求可能会解决该问题。
503	None	503 响应

OPTIONS

通过返回正确标头来启用 CORS

路径参数

名称	Type	必需	描述
ReplicatorArn	String	True	要描述的 MSK 复制器的Amazon 资源名称（ARN）。

响应

状态代码	响应模型	描述
200	None	CORS 方法的默认响应。

架构

请求正文

PUT 架构

```
{
  "consumerGroupReplication": {
    "consumerGroupsToExclude": [
      "string"
    ],
    "detectAndCopyNewConsumerGroups": boolean,
    "consumerGroupsToReplicate": [
      "string"
    ],
    "synchroniseConsumerGroupOffsets": boolean
  },
  "topicReplication": {
    "copyAccessControlListsForTopics": boolean,
    "detectAndCopyNewTopics": boolean,
    "copyTopicConfigurations": boolean,
    "topicsToReplicate": [
      "string"
    ],
    "topicsToExclude": [
      "string"
    ]
  },
  "sourceKafkaClusterArn": "string",
  "targetKafkaClusterArn": "string",
  "currentVersion": "string"
}
```

响应正文

UpdateReplicationInfoResponse 架构

```
{
  "replicatorArn": "string",
  "replicatorState": enum
}
```

属性

ConsumerGroupReplicationUpdate

有关使用器组复制的详细信息。

consumerGroupsToExclude

指定不应复制的使用器组的正则表达式模式列表。

类型：string 类型的数组
必需：True
MaxLength：256

detectAndCopyNewConsumerGroups

启用使用器组与 MSK 复制器目标集群的同步。

类型：布尔值
必需：True

consumerGroupsToReplicate

指定要复制的使用器组的正则表达式模式列表。

类型：string 类型的数组
必需：True
MaxLength：256

synchroniseConsumerGroupOffsets

启用使用器组偏移与 MSK 复制器目标集群的同步。转换后的偏移将写入主题 __consumer_offsets。

类型：布尔值

必需：True

ReplicatorState

MSK 复制器的状态。

RUNNING
CREATING
UPDATING
DELETING
FAILED

TopicReplicationUpdate

用于更新 MSK 复制器的主题复制的详细信息。

copyAccessControlListsForTopics

是否定期配置远程主题 ACL 以匹配其对应的上游主题。

类型：布尔值

必需：True

detectAndCopyNewTopics

是否定期检查新主题和分区。

类型：布尔值

必需：True

copyTopicConfigurations

是否定期配置远程主题以匹配其对应的上游主题。

类型：布尔值

必需：True

topicsToReplicate

指定要复制的主题的正则表达式模式列表。

类型：string 类型的数组
必需：True
MaxLength: 249

topicsToExclude

指定不应复制的主题的正则表达式模式列表。

类型：string 类型的数组
必需：True
MaxLength: 249

UpdateReplicationInfoRequest

用于更新 MSK 复制器的源和目标 Kafka 集群之间复制信息的参数。

consumerGroupReplication

更新了使用器组复制信息。

类型:ConsumerGroupReplicationUpdate
必填项：False

topicReplication

更新了主题复制信息。

类型:TopicReplicationUpdate
必填项：False

sourceKafkaClusterArn

MSK 复制器源 Kafka 集群的 Amazon 资源名称（ARN）。

类型：字符串
必需：True

targetKafkaClusterArn

MSK 复制器目标 Kafka 集群的 Amazon 资源名称（ARN）。

类型：字符串
必需：True

currentVersion

MSK 复制器的当前版本。

类型：字符串
必需：True

UpdateReplicationInfoResponse

更新了 MSK 复制器的复制信息。

replicatorArn

MSK 复制器的 Amazon 资源名称（ARN）。

类型：字符串
必填项：False

replicatorState

MSK 复制器的状态。

类型:ReplicatorState
必填项：False

另请参阅

有关在特定语言的 AWS SDK 和参考中使用此 API 的更多信息，请参阅以下内容：

UpdateReplicationInfo

- AWS 命令行界面

- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS 适用于 JavaScript
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包
- AWS 适用于 Ruby V3 的 SDK

《Amazon MSK 开发人员指南》文档历史记录

下表介绍了对《Amazon MSK 开发人员指南》的重要更改。

最新文档更新：2024 年 2 月 7 日

更改	描述	日期
`WriteDataIdempotently` 已添加到 AWSMSKReplicatorExecutionRole	`WriteDataIdempotently` 权限已添加到 AWSMSKReplicatorExecutionRole 策略中，以支持 MSK 集群之间的数据复制。	2024-3-12
Graviton M7g经纪商在巴西和巴林上市。	Amazon MSK现在支持使用G AWS raviton处理器（由亚马逊网络服务构建的基于ARM的定制处理器）的m7g经纪商在南美（sa-east-1，圣保罗）和中东（me-south-1，巴林）地区的可用性。	2024-2-07
向中国地区释放 Graviton m7g 经纪商	亚马逊 MSK 现在支持使用 G AWS raviton 处理器（由亚马逊网络服务构建的基于 ARM 的定制 ARM 处理器）的 m7g 经纪商在中国地区上市。	2024-01-11
亚马逊 MSK Kafka 版本支持政策	增加了对亚马逊 MSK 支持的 Kafka 版本支持政策的解释。有关更多信息，请参阅 Apache Kafka 版本。	2023-12-08
支持 Amazon MSK Replicator 的新服务执行角色策略。	亚马逊 MSK 添加了支持亚马逊 MSK Replicator 的新`AWSMSKReplicatorExecutionRole` 政策。有关更多信息，请参阅 AWS	2023-12-06

更改	描述	日期
	托管策略：AWSMSKReplicatorExecutionRole 。	
m7g Graviton 支持	亚马逊 MSK 现在支持使用 G AWS raviton 处理器（由 Amazon Web Services 构建的基于 ARM 的定制处理器）的 m7g 代理。	2023-11-27
Amazon MSK 复制器	Amazon MSK 复制器是一项新功能，可用于在 Amazon MSK 集群之间复制数据。亚马逊 MSK Replicator 包括对 Amazon FullAccess MSK 政策的更新。有关更多信息，请参阅 AWS 托管策略：AmazonMSKFullAccess 。	2023-09-28
更新 IAM 最佳实践。	更新了指南，使其符合 IAM 最佳实践。有关更多信息，请参阅 IAM 安全最佳实践。	2023-03-08

更改	描述	日期
服务相关角色更新为支持多 VPC 私有连接	Amazon MSK 现在包含 AWSServiceRoleForKafka 服务相关角色更新，用于管理您账户中的网络接口和 VPC 终端节点，从而使您的 VPC 中的客户可以访问集群代理。Amazon MSK 对 `DescribeVpcEndpoints` 、`ModifyVpcEndpoint` 和 `DeleteVpcEndpoints` 使用权限。有关更多信息，请参阅 对 Amazon MSK 使用服务相关角色。	2023-03-08
支持 Apache Kafka 2.7.2	Amazon MSK 现在支持 Apache Kafka 版本 2.7.2。有关更多信息，请参阅 支持的 Apache Kafka 版本。	2021-12-21
支持 Apache Kafka 2.6.3	Amazon MSK 现在支持 Apache Kafka 版本 2.6.3。有关更多信息，请参阅 支持的 Apache Kafka 版本。	2021-12-21
MSK Serverless 预发行	MSK Serverless 是一项新功能，可用于创建无服务器集群。有关更多信息，请参阅 MSK Serverless。	2021-11-29
支持 Apache Kafka 2.8.1	Amazon MSK 现在支持 Apache Kafka 版本 2.8.1。有关更多信息，请参阅 支持的 Apache Kafka 版本。	2021-09-30

更改	描述	日期
MSK Connect	MSK Connect 是一项新功能，可用于创建和管理 Apache Kafka 连接器。有关更多信息，请参阅 MSK Connect。	2021-09-16
支持 Apache Kafka 2.7.1	Amazon MSK 现在支持 Apache Kafka 版本 2.7.1。有关更多信息，请参阅 支持的 Apache Kafka 版本。	2021-05-25
支持 Apache Kafka 2.8.0	Amazon MSK 现在支持 Apache Kafka 版本 2.8.0。有关更多信息，请参阅 支持的 Apache Kafka 版本。	2021-04-28
支持 Apache Kafka 2.6.2	Amazon MSK 现在支持 Apache Kafka 版本 2.6.2。有关更多信息，请参阅 支持的 Apache Kafka 版本。	2021-04-28
支持更新代理类型	现在，您可以更改现有集群的代理类型。有关更多信息，请参阅 更新代理类型。	2021-01-21
支持 Apache Kafka 2.6.1	Amazon MSK 现在支持 Apache Kafka 版本 2.6.1。有关更多信息，请参阅 支持的 Apache Kafka 版本。	2021-01-19
支持 Apache Kafka 2.7.0	Amazon MSK 现在支持 Apache Kafka 版本 2.7.0。有关更多信息，请参阅 支持的 Apache Kafka 版本。	2020-12-29

更改	描述	日期
Apache Kafka 版本 1.1.1 将无法创建新集群	您无法再使用 Apache Kafka 版本 1.1.1 创建新的 Amazon MSK 集群。但是，如果您有运行 Apache Kafka 版本 1.1.1 的现有 MSK 集群，则可以继续在这些现有集群上使用当前支持的所有功能。有关更多信息，请参阅 Apache Kafka 版本。	2020-11-24
使用器滞后指标	Amazon MSK 现在提供可用来监控使用器滞后的指标。有关更多信息，请参阅 监控 Amazon MSK 集群。	2020-11-23
支持 Cruise Control	亚马逊 MSK 现在支持巡 LinkedIn航控制系统。有关更多信息，请参阅 将 Amazon MSK 与 LinkedIn 的适用于 Apache Kafka 的 Cruise Control 结合使用。	2020-11-17
支持 Apache Kafka 2.6.0	Amazon MSK 现在支持 Apache Kafka 版本 2.6.0。有关更多信息，请参阅 支持的 Apache Kafka 版本。	2020-10-21
支持 Apache Kafka 2.5.1	Amazon MSK 现在支持 Apache Kafka 版本 2.5.1。在 Apache Kafka 版本 2.5.1 中，亚马逊 MSK 支持在客户端和终端节点之间传输时进行加密。 ZooKeeper 有关更多信息，请参阅 支持的 Apache Kafka 版本。	2020-09-30

更改	描述	日期
应用程序自动扩展	您可以配置 Amazon Managed Streaming for Apache Kafka，使其在使用量增加时自动扩展集群的存储空间。有关更多信息，请参阅 自动扩缩。	2020-09-30
支持用户名和密码安全	Amazon MSK 现在支持使用用户名和密码登录集群。亚马逊 MSK 将凭证存储在 S AWS ecrets Manager 中。有关更多信息，请参阅 SASL/SCRAM 身份验证。	2020-09-17
支持升级 Amazon MSK 集群的 Apache Kafka 版本	您现在可以升级现有 MSK 集群的 Apache Kafka 版本。	2020-05-28
支持 T3.small 代理节点	Amazon MSK 现在支持使用 Amazon EC2 类型 T3.small 代理创建集群。	2020-04-08
支持 Apache Kafka 2.4.1	Amazon MSK 现在支持 Apache Kafka 版本 2.4.1。	2020-04-02
支持流式传输代理日志	亚马逊 MSK 现在可以将代理日志流式传输到 CloudWatch Logs、Amazon S3 和 Amazon Data Firehose。反过来，Firehose 可以将这些日志传送到其支持的目的地，例如 OpenSearch 服务。	2020-02-25
支持 Apache Kafka 2.3.1	Amazon MSK 现在支持 Apache Kafka 版本 2.3.1。	2019-12-19
开源监控系统	Amazon MSK 现在支持 Prometheus 的开源监控系统。	2019-12-04

更改	描述	日期
支持 Apache Kafka 2.2.1	Amazon MSK 现在支持 Apache Kafka 版本 2.2.1。	2019-07-31
公开发行	新功能包括标记支持、身份验证、TLS 加密、配置以及代理存储更新功能。	2019-05-30
支持 Apache Kafka 2.1.0	Amazon MSK 现在支持 Apache Kafka 版本 2.1.0。	2019-02-05

AWS 词汇表

有关最新的 AWS 术语，请参阅《AWS 词汇表参考》中的 AWS 词汇表。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。