aws

開發人員指南

Amazon Managed Streaming for Apache Kafka

Amazon Managed Streaming for Apache Kafka: 開發人員指南

Table of Contents

歡迎使用《Amazon MSK 開發人員指南》

歡迎使用《Amazon MSK 開發人員指南》。下列主題可根據您要執行的工作，協助您開始使用本指南。

- 依照 開始使用 Amazon MSK 教學課程建立 Amazon MSK 叢集。
- 參閱 Amazon MSK：運作方式 可深入了解 Amazon MSK 的功能。
- 使用 MSK Serverless 可執行 Apache Kafka，無須管理和擴展叢集容量。
- 使用 MSK Connect 可將資料串流到或串流出 Apache Kafka 叢集。
- 用MSK Replicator於在不同或相同 AWS 區域的 Amazon MSK 叢集之間可靠地複寫資料。

如需重點資訊、產品詳細資訊和定價資訊，請參閱 Amazon MSK 的服務頁面。

什麼是 Amazon MSK？

Amazon Managed Streaming for Apache Kafka (Amazon MSK) 是一項全受管服務，可讓您建置和執行使用 Apache Kafka 處理串流資料的應用程式。Amazon MSK 提供控制平面操作，例如用於建立、更新和刪除叢集的操作。它可以讓你使用 Apache Kafka 資料平面操作，如那些用於生產和使用數據。其執行 Apache Kafka 的開源版本。這表示支援現有的應用程式，工具以及合作夥伴和 Apache Kafka 社群的外掛程式，而無須變更應用程式代碼。您可使用 Amazon MSK 建立使用 the section called “支援的 Apache Kafka 版本” 章節所列任一 Apache Kafka 版本的叢集。

這些組件描述了 Amazon MSK 的架構：

- 代理程式節點 – 建立 Amazon MSK 叢集時，您可指定想要 Amazon MSK 在每個可用區域中建立的代理程式節點數量。每個可用區域最少為一個代理程式。每個可用區域都有自己的虛擬私有雲端 (VPC) 子網路。
- ZooKeeper 節點 — Amazon MSK 也會為您建立 Apache ZooKeeper 節點。Apache ZooKeeper 是一個開放原始碼伺服器，可實現高度可靠的分散式
- 卡夫卡控制器-阿帕奇卡夫卡社區開發卡夫卡取代 Apache ZooKeeper 的元數據管理在阿帕奇卡夫卡集群. 在 Kraft 模式下，集群元數據是一組卡夫卡控制器，這是卡夫卡集群的一部分，而不是跨節點內傳播。 ZooKeeper 隨附 Kraft 控制器，無需額外付費，無需額外的設定或管理。

> **Note**
>
> 從阿帕奇卡夫卡 3.7.x 版本 MSK，您可以創建使用卡夫模式，而不是模式集群。 ZooKeeper

- 生產者，取用者和主題建立者 – Amazon Msk 可讓您使用 Apache Kafka 資料平面操作來建立主題，以及生產和取用資料。
- 叢集作業您可以 AWS Management Console使用 SDK 中的、 AWS Command Line Interface (AWS CLI) 或 API 來執行控制平面作業。例如，您可以建立或刪除 Amazon MSK 叢集、列出帳戶中的所有叢集、檢視叢集的屬性，以及更新叢集中代理程式的數量和類型。

Amazon MSK 會偵測叢集最常見的故障案例並自動復原，以便您的生產者和取用者應用程式能夠在影響最小的情況下繼續寫入和讀取操作。當 Amazon MSK 偵測到代理程式故障時，其會緩解故障，或是使用新的代理程式來取代運作狀態不良或是無法連線的代理程式。除此之外，在可能的情況下，它重用從舊的代理程式的儲存體，以減少 Apache Kafka 需要複製的資料。您受到可用性影響時間將會僅限於 Amazon MSK 完成偵測與復原所需的時間。復原之後，您的生產者和取用者應用程式可以繼續與失敗前所使用的相同代理 IP 地址進行通訊。

設定 Amazon MSK

首次使用 Amazon MSK 之前，請先完成下列任務：

任務

- 註冊成為 AWS
- 下載程式庫和工具

註冊成為 AWS

當您註冊時 AWS，您的 Amazon Web Services 帳戶會自動註冊所有服務 AWS，包括 Amazon MSK。您只需支付實際使用服務的費用。

如果您已經有 AWS 帳號，請跳至下一個工作。若您尚未擁有 AWS 帳戶，請使用下列程序建立帳戶。

註冊 Amazon Web Services 帳戶

1. 開啟 https://portal.aws.amazon.com/billing/signup。
2. 請遵循線上指示進行。

   部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

   當您註冊一個時 AWS 帳戶，將創建AWS 帳戶根使用者一個。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。安全性最佳做法是將管理存取權指派給使用者，並僅使用 root 使用者來執行需要 root 使用者存取權的工作。

下載程式庫和工具

下列程式庫和工具可協助您使用 Amazon MSK：

- AWS Command Line Interface (AWS CLI) 支援 Amazon MSK。 AWS CLI 可讓您透過命令列控制多個 Amazon Web Services，並透過指令碼將它們自動化。升級 AWS CLI 到最新版本，以確保其支援本使用者指南中所述的 Amazon MSK 功能。如需有關 AWS CLI升級方式的詳細指示，請參閱安裝 AWS Command Line Interface。安裝之後 AWS CLI，您必須對其進行配置。有關如何設定的資訊 AWS CLI，請參閱 aws 設定。
- Amazon Managed Streaming for Kafka API Reference 記錄了 Amazon MSK 支援的 API 操作。

- 適用於圍棋、Java、.NET、Node.js、PHP、JavaScriptPython 和紅寶石的亞馬 Amazon Web Services 開發套件包括 Amazon MSK 支援和範例。

開始使用 Amazon MSK

本教學課程將示範如何建立 MSK 叢集、產生和取用資料，以及使用指標監控叢集的運作狀態。此範例不代表在您建立 MSK 叢集時可選擇的所有選項。在本教學的不同部分中，為求簡化我們選擇預設選項。這並不表示只有這些選項才能用於設定 MSK 叢集或用戶端執行個體。

主題

步驟 1：建立 Amazon MSK 叢集

在開始使用 Amazon MSK 的這個步驟中，您會建立一個 Amazon MSK 叢集。

若 Amazon 使用 AWS Management Console

1. 登入 AWS Management Console，然後開啟 Amazon MSK 主控台，網址為 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。
2. 選擇建立叢集。
3. 針對建立方法，請確認已選取快速建立選項。快速建立選項可讓您使用預設設定建立叢集。
4. 針對叢集名稱，輸入叢集的描述性名稱。例如 **`MSKTutorialCluster`**。
5. 針對一般叢集屬性，請選擇佈建作為叢集類型。
6. 從所有叢集設定下的表格中，複製下列設定值並儲存，供在本教學課程中稍後使用：
   - VPC
   - 子網
   - 與 VPC 關聯的安全群組
7. 選擇建立叢集。

8. 在叢集摘要頁面查看叢集狀態。當 Amazon MSK 佈建叢集時，叢集狀態會從建立變更為作用中。叢集狀態為作用中時，您可以連線至叢集。如需有關叢集狀態的詳細資訊，請參閱叢集狀態。

後續步驟

步驟 2：建立 IAM 角色

步驟 2：建立 IAM 角色

在此步驟中，您會執行兩項任務。第一項任務是建立 IAM 政策，用於授予在叢集上建立主題，並將資料傳送至這些主題的存取權限。第二項任務是建立 IAM 角色，並將此政策與該角色建立關聯。在稍後的步驟中，您會建立擔任此角色的用戶端機器，使用它在叢集上建立主題，並將資料傳送至該主題。

建立能夠建立和寫入主題的 IAM 政策

1. 開啟位於 https://console.aws.amazon.com/iam/ 的 IAM 主控台。
2. 在導覽窗格中，選擇政策。
3. 選擇建立政策。
4. 選擇 JSON 索引標籤，然後使用下列 JSON 取代編輯器視窗中的 JSON。

   *######## AWS ###########*使用您的帳戶 ID 取代 *Account-ID*。TutorialCluster以叢集的名稱取代 *MSK*。

```
{
    "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/*"
            ]
        }
    ]
}
```

如需有關安全政策撰寫方式的相關說明，請參閱 the section called “IAM 存取控制”。

5. 選擇下一步：標籤。
6. 選擇下一步：檢閱。
7. 針對政策名稱，請輸入描述性名稱，例如 msk-tutorial-policy。
8. 選擇建立政策。

建立 IAM 角色，並將政策連接至該角色

1. 在導覽窗格中，選擇角色。
2. 選擇建立角色。
3. 在一般使用案例下，選擇 EC2，然後選擇下一步：許可。
4. 在搜尋方塊中，輸入您先前為此教學課程建立的政策名稱。然後選取政策左側的核取方塊。
5. 選擇下一步：標籤。
6. 選擇下一步：檢閱。
7. 針對角色名稱，請輸入描述性名稱，例如 msk-tutorial-role。
8. 選擇建立角色。

後續步驟

步驟 3：建立用戶端機器

步驟 3：建立用戶端機器

在開始使用 Amazon MSK 的這個步驟中，您會建立用戶端機器。您可以使用此用戶端機器來建立產生和取用資料的主題。為了簡化起見，您將在與 MSK 叢集相關聯的 VPC 中建立用戶端機器，方便用戶端輕鬆連線到叢集。

建立用戶端機器

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 選擇啟動執行個體。
3. 輸入用戶端機器的名稱，例如 **MSKTutorialClient**。
4. 確認已選擇 Amazon Linux 2 AMI (HVM) – Kernel 5.10，SSD 磁碟區類型作為 Amazon Machine Image (AMI) 類型。
5. 確認選取 t2.micro 執行個體類型。
6. 在金鑰對 (登入) 下，選擇建立新金鑰對。輸入 **MSKKeyPair** 作為金鑰對名稱，然後選擇下載金鑰對。或者，您也可以使用現有的金鑰對。
7. 展開進階詳細資料區段，然後選擇您在步驟 2：建立 IAM 角色中建立的 IAM 角色。
8. 選擇啟動執行個體。
9. 選擇檢視執行個體。然後，在安全群組資料欄中，選擇與新執行個體相關聯的安全群組。複製並儲存安全群組的 ID，以供日後使用。
10. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
11. 在導覽窗格中，選擇安全群組。尋找您儲存在 the section called “步驟 1：建立叢集” 中的安全群組 ID。
12. 在傳入規則索引標籤中，選擇編輯傳入規則。
13. 選擇新增規則。
14. 在新規則中，於類型資料欄中選擇所有流量。在來源資料欄的第二個欄位中，選擇用戶端機器的安全群組。這是您在啟動用戶端機器執行個體之後儲存的群組名稱。
15. 選擇儲存規則。現在，叢集的安全群組就能接受來自用戶端機器安全群組的流量了。

後續步驟

步驟 4：建立主題

步驟 4：建立主題

在開始使用 Amazon MSK 的這個步驟中，您可以在用戶端機器上安裝 Apache Kafka 用戶端程式庫和工具，然後建立主題。

> **Warning**
>
> 本教學課程中使用的 Apache Kafka 版本號僅為示例。我們建議您使用與 MSK 叢集版本相同的用戶端版本。較舊的用戶端版本可能缺少特定功能和重大錯誤修正。

尋找 MSK 叢集的版本

1. 移至 https://eu-west-2.console.aws.amazon.com/msk/
2. 選取 MSK 叢集。
3. 請記住叢集上使用的 Apache Kafka 版本。
4. 使用步驟 3 中取得的版本取代本教學課程中 Amazon MSK 版本號碼的執行個體。

若要在用戶端機器上建立主題

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。然後選取您在 步驟 3：建立用戶端機器 建立的用戶端機器名稱旁邊的核取方塊。
3. 選擇動作，然後選擇連線。遵循主控台的指示操作，連線至您的用戶端機器。
4. 執行下列命令，在用戶端機器上安裝 Java：

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

5. 執行下列命令下載 Apache Kafka。

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

> **Note**
>
> 如果您想要使用此命令中以外的鏡像網站，您可以在 Apache 網站上選擇不同的鏡像網站。

6. 在您在先前步驟中下載 TAR 檔案的目錄中執行下列命令。

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

7. 前往 `kafka_2.13-{YOUR MSK VERSION}/libs` 目錄，然後執行下列命令以下載 Amazon MSK IAM JAR 檔案。Amazon MSK IAM JAR 可讓用戶端機器存取叢集。

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

8. 前往 `kafka_2.13-{YOUR MSK VERSION}/bin` 目錄。複製下列屬性設定，並將其貼入新檔案。將檔案命名為 **`client.properties`** 並儲存。

```
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. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
10. 等待叢集的狀態變成作用中。這可能需要幾分鐘的時間。狀態變為作用中之後，選擇叢集名稱。這會帶您前往包含叢集摘要的頁面。
11. 選擇檢視用戶端資訊。
12. 複製私有端點的連線字串。

    您的每個代理程式將獲得三個端點。您只需要一個代理程式端點即可執行下列步驟。
13. 執行下列命令，將 *`BootstrapServerString`* 取代為您在上一個步驟中取得的其中一個代理程式端點。

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

    如果命令成功，您會看到以下訊息：`Created topic MSKTutorialTopic.`

後續步驟

步驟 5：產生和取用資料

步驟 5：產生和取用資料

在開始使用 Amazon MSK 的這個步驟中，您會產生和取用資料。

產生和取用訊息

1. 執行下列命令以啟動主控台生產者。*###########BootstrapServer############ String#*如需有關如何擷取此連線字串的指示，請參閱取得 Amazon MSK 叢集的引導代理程式。

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

2. 輸入您想要的任何訊息，然後按 Enter 鍵。重複此步驟兩次或三次。每次輸入一行，然後按 Enter，該行會作為單獨訊息傳送到您的 Apache Kafka 叢集中。
3. 保持與用戶端機器的連線開啟，然後開啟第二個並在新視窗中與該機器單獨連線。
4. *########BootstrapServer############### String#*然後，若要建立主控台取用者，請執行下列命令,將第二個連線連接至用戶端機器。

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

   當您使用主控台生產者命令時，您會開始看到先前輸入的訊息。
5. 在生產者視窗中輸入更多訊息，並觀看它們出現在取用者視窗中。

後續步驟

步驟 6：使用 Amazon CloudWatch 查看 Amazon MSK 指標

步驟 6：使用 Amazon CloudWatch 查看 Amazon MSK 指標

在開始使用 Amazon MSK 的這一步中，您可以查看 Amazon 中的 Amazon MSK 指標。 CloudWatch

若要在中檢視 Amazon MSK 指標 CloudWatch

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇指標。
3. 選擇所有指標索引標籤，然後選擇 AWS/Kafka。
4. 若要檢視代理程式層級指標，請選擇代理程式 ID，叢集名稱。針對叢集層級指標，請選擇叢集名稱。
5. (選擇性) 在圖表窗格中，選取統計值和期間，然後使用這些設定建立 CloudWatch 警示。

後續步驟

步驟 7：刪除為此教學課程建立的 AWS 資源

步驟 7：刪除為此教學課程建立的 AWS 資源

在開始使用 Amazon MSK 的最後一個步驟中，您將刪除在本教學課程中建立的 MSK 叢集和用戶端機器。

若要使用刪除資源 AWS Management Console

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
2. 選取叢集的名稱。例如，MSK TutorialCluster。
3. 選擇動作，然後選擇刪除。
4. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
5. 選擇您為用戶端機器建立的執行個體，例如 **MSKTutorialClient**。
6. 依序選擇執行個體狀態和終止執行個體。

刪除 IAM 政策和角色

1. 前往 https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。
2. 在導覽窗格中，選擇角色。
3. 在搜尋方塊中，輸入您為本教學課程建立的 IAM 角色名稱。
4. 選擇角色。然後選擇刪除角色並確認刪除。
5. 在導覽窗格中，選擇政策。
6. 在搜尋方塊中，輸入您為本教學課程建立的政策名稱。

7. 選擇政策以開啟其摘要頁面。在政策的摘要頁面上，選擇刪除政策。
8. 選擇刪除。

Amazon MSK：運作方式

Amazon MSK 叢集是您可以在帳戶中建立的主要 Amazon MSK 資源。本節中的主題說明如何執行常見的 Amazon MSK 操作。如需可在 MSK 叢集上執行的所有操作清單，請參閱下列內容：

- AWS Management Console
- Amazon MSK API Reference
- Amazon MSK CLI Command Reference

主題

建立 Amazon MSK 叢集

> **Important**
>
> 在您建立叢集後，無法變更 Amazon MSK 叢集的 VPC。

您需要先擁有 Amazon Virtual Private Cloud (VPC) 並在該 VPC 內設定子網路，才能建立 Amazon MSK 叢集。

您需要有兩個子網路，分別在美國西部 (加利佛尼亞北部) 區域的兩個不同可用區域中。在可使用 Amazon MSK 的所有其他區域，您可以指定兩個或三個子網路。您的子網路必須位於不同的可用區域。建立叢集時，Amazon MSK 會將代理程式節點平均分配到您指定的子網路。

經紀人規模

建立 Amazon MSK 叢集時，您可以指定所要的代理程式大小。Amazon MSK 支援下列代理程式大小：

- kafka.t3.small
- kafka.m5.large、kafka.m5.xlarge、kafka.m5.2xlarge、kafka.m5.4xlarge、kafka.m5.8xlarge、kafka.m5.12x
- 卡夫卡 .m7 克大, 卡夫卡 .m7 公克大, 卡夫卡 .m7 公克大, 卡夫卡 .m7 公克大, 卡夫卡 .m7 公克大, 卡夫卡 .m7 公克大, 12 倍大, 卡夫卡 .m7 公克

M7g 代理程式使用 AWS 重力發處理器 (由 Amazon Web Services 建置的自訂 ARM 處理器)。與同類 M5 實例相比，M7g 經紀人提供了改善的價格性能。與同類 M5 執行個體相比，M7g 代理程式消耗的電力更少。

M7g 重力頓代理商不在以下地區提供：CDG（巴黎），CGK（雅加達），CPT（開普敦），DXB（杜拜），HKG（香港），KIX（大阪），倫敦 LHR（倫敦），MEL（墨爾本），MXP（米蘭），OSU（美國東部），太平洋時間（美國西部），ZLV（卡爾加里），特拉維夫（YYC）RH（蘇黎世）。

MSK 在執行以下卡夫卡版本之一的叢集上支援 M7g 代理程式：

- 2. 分層
- 3.3.2
- 3.4.0
- 3.5.1
- 3.6.0 含階層式儲存
- 3.7.x
- 3.7.x. 卡夫

與 T3 代理程式相比，M7g 和 M5 代理程式具有更高的基準輸送量效能，建議用於生產工作負載。與 T3 經紀人相比，M7g 和 M5 經紀人每個代理人還可以擁有更多的分區。如果您正在執行較大的生產級工作負載或需要更多分割區，請使用 M7g 或 M5 代理程式。若要進一步了解有關 M7g 和 M5 執行個體大小的資訊，請參閱 Amazon EC2 一般用途執行個體。

T3 中介裝置有能力使用 CPU 信用來暫時爆增性能。如果您正在測試中小型串流工作負載，或如果您有低輸送量串流工作負載，會遇到暫時性尖峰的輸送量，請使用 T3 中介裝置進行低成本開發。我們建議您執行 proof-of-concept 測試，以判斷 T3 代理程式是否足以應付生產環境或關鍵工作負載。若要進一步了解 T3 代理程式規模，請參閱 Amazon EC2T3 執行個體。

有關如何選擇經紀人規模的更多信息，請參閱最佳實務。

使用建立叢集 AWS Management Console

此程序說明使用自訂建立選項建立已佈建叢集的常見工作。您可以在 MSK 主控台中選取其他選項來建立無伺服器叢集。

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
2. 選擇建立叢集。
3. 針對叢集建立方法，選擇 [自訂建立]。
4. 指定唯一且不超過 64 個字元的叢集名稱。
5. 對於叢集類型，請選擇已佈建，您可以指定代理程式數目、代理程式大小和叢集儲存容量。
6. 選擇要在經紀人上運行的 Apache 卡夫卡版本。若要查看每個 Apache Kafka 版本支援的 MSK 功能比較，請選取 [檢視版本相容性]。
7. 視您選取的 Apache Kafka 版本而定，您可以選擇叢集的中繼資料模式：ZooKeeper 或 Kraft。
8. 根據叢集的運算、記憶體和儲存區需求，選取要用於叢集的代理程式大小。請參閱???，
9. 選取分配代理程式的區域數目。
10. 指定您希望 MSK 在每個可用區域中建立的代理程式數量。每個可用區域最少為一個代理程式，而針對基礎叢集，每個叢集最多為 30 個代理程 ZooKeeper式，KRAF 型叢集的每個叢集最多為 60 個代理程式。
11. 選取您希望叢集擁有的初始儲存容量。建立叢集之後，就無法減少儲存容量。
12. 根據您選取的代理程式大小 (執行個體大小)，您可以指定每個代理程式的佈建儲存輸送量。若要啟用此選項，請針對 x86 選擇代理程式大小 (執行個體大小) kafka.m5.4xlarge 或更大，對於以重力為基礎的執行個體，請選擇 kafka.m7g.2xlarge 或更大。請參閱???。
13. 選取叢集儲存區模式選項，可以是僅 EBS 儲存體或階層式儲存體和 EBS 儲存區。

14. 如果您想要建立並使用自訂叢集配置 (或者如果您已經儲存叢集配置)，請選擇一個配置。否則，您可以使用 Amazon MSK 預設叢集組態建立叢集。如需有關 Amazon MSK 組態的資訊，請參閱 組態。
15. 選取下一步。
16. 對於網路設定，請選擇要用於叢集的 VPC。
17. 根據您先前選取的區域數目，指定要部署代理程式的可用區域和子網路。子網路必須位於不同的可用區域中。
18. 您可以選取一或多個要授予叢集存取權的安全性群組 (例如，用戶端機器的安全性群組)。如果您指定與您共用的安全性群組，則必須確定您擁有使用這些群組的權限。具體而言，您需要 `ec2:DescribeSecurityGroups` 許可。連線至 Amazon MSK 叢集。
19. 選取下一步。
20. 選取叢集的存取控制方法和加密設定，以便在用戶端和代理程式之間傳輸時加密資料。如需詳細資訊，請參閱 the section called “傳輸中加密”。
21. 選擇您要用來加密靜態資料的 KMS 金鑰。如需詳細資訊，請參閱 the section called “靜態加密”。
22. 選取下一步。
23. 選擇您想要的監控和標籤。這會決定您取得的指標組。如需詳細資訊，請參閱 監控叢集。Amazon CloudWatch 、Prometheus、代理程式記錄傳送或叢集標籤，然後選取 [下一步]。
24. 檢閱叢集的設定。您可以返回並變更設定，方法是選取 [上一步] 返回上一個主控台畫面，或選取 [編輯] 以變更特定叢集設定。如果設定正確，請選取 [建立叢集]。
25. 在叢集摘要頁面查看叢集狀態。當 Amazon MSK 佈建叢集時，叢集狀態會從建立變更為作用中。叢集狀態為作用中時，您可以連線至叢集。如需有關叢集狀態的詳細資訊，請參閱叢集狀態。

使用建立叢集 AWS CLI

1. 複製以下 JSON 並將其儲存到檔案。將檔案命名為 `brokernodegroupinfo.json`。將 JSON 中的子網路 ID 取代為與子網路對應的值。這些子網路必須位於不同的可用區域。將 *"Security-Group-ID"* 取代為用戶端 VPC 的一或多個安全群組 ID。與這些安全群組關聯的用戶端會獲得叢集的存取權。如果指定已與自己共用的安全群組，您必須確保自己擁有這些群組的許可。具體而言，您需要 `ec2:DescribeSecurityGroups` 許可。如需範例，請參閱 Amazon EC2：允許以程式設計方式和在主控台中管理具備特定標籤鍵值對的 EC2 安全群組。最後，將更新的 JSON 文件保存在已 AWS CLI 安裝的計算機上。

```
{
  "InstanceType": "kafka.m5.large",
```

```
  "ClientSubnets": [
    "Subnet-1-ID",
    "Subnet-2-ID"
  ],
  "SecurityGroups": [
    "Security-Group-ID"
  ]
}
```

> **Important**
>
> 如果您正在使用美國西部 (加州北部) 區域，請精確指定兩個子網路。針對其他可使用 Amazon MSK 的區域，您可以指定兩個或三個子網路。您指定的子網路必須位於不同的可用區域。建立叢集時，Amazon MSK 會將中繼節點平均分散到您指定的子網路。

2. 在保存brokernodegroupinfo.json文件的目錄中運行以下 AWS CLI 命令，將 *########* 替換為您選擇的名稱。對於 *"Monitoring-Level" (######)*，您可以指定下列三個值之一：DEFAULT、PER_BROKER 或 PER_TOPIC_PER_BROKER。如需這三種不同監控層級的相關資訊，請參閱 ???。enhanced-monitoring 為選用參數。如果您沒有在 create-cluster 命令中進行指定，則會獲得 DEFAULT 監控層級。

```
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"
```

命令的輸出如下 JSON 所示：

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

> **Note**
>
> create-cluster 命令可能會傳回錯誤，指出一或多個子網路屬於不支援的可用區域。發生這種情況時，這個錯誤會指出哪些可用區域不受支援。建立子網路，該子網路不使用不受支援的可用區域，然後再試一次 create-cluster 命令。

3. 儲存 ClusterArn 金鑰的值，因為您需要這個資訊來在叢集上執行其他動作。
4. 執行下列命令，以檢查您的叢集 STATE。當 Amazon MSK 佈建叢集時，STATE 值會從 CREATING 變更為 ACTIVE。叢集狀態為 ACTIVE 時，您可以連線至叢集。如需有關叢集狀態的詳細資訊，請參閱叢集狀態。

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

使用建立具有自訂 Amazon MSK 組態的叢集 AWS CLI

如需有關自訂 Amazon MSK 組態及如何進行建立該組態的資訊，請參閱 組態。

1. 將以下 JSON 儲存到檔案中，使用您要用來建立叢集的組態 ARN 取代 *configuration-arn*。

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

2. 執行 create-cluster 命令並使用 configuration-info 選項來指向您在上一步中儲存的 JSON 檔案。以下是範例。

```
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
```

以下是執行此命令後成功回應的範例。

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

使用 API 建立叢集

若要使用 API 建立叢集，請參閱CreateCluster。

刪除 Amazon MSK 叢集

> **Note**
>
> 如果您的叢集具有自動擴展政策，建議您先移除該政策，再刪除叢集。如需詳細資訊，請參閱自動調整規模。

使用刪除叢集 AWS Management Console

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
2. 選取您要刪除的 MSK 叢集旁的核取方塊來選擇該叢集。
3. 選擇刪除，並確認刪除。

使用刪除叢集 AWS CLI

執行下列命令，取代*ClusterArn*為建立叢集時取得的 Amazon 資源名稱 (ARN)。若您沒有叢集的 ARN，可透過列出所有叢集來找到該 ARN。如需詳細資訊，請參閱 the section called “列出叢集”。

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

使用 API 刪除叢集

若要使用 API 刪除叢集，請參閱DeleteCluster。

取得 Amazon MSK 叢集的引導代理程式

使用獲取引導代理 AWS Management Console

引導代理程式這一術語，是指 Apache Kafka 用戶端可以用作連線至叢集之起始點的代理程式清單。此清單不一定包含叢集中的所有代理程式。

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。

2. 此表格會顯示此帳戶下目前區域的所有叢集。選擇叢集名稱以檢視其描述。
3. 在叢集摘要頁面上，選擇檢視用戶端資訊。這將顯示啟動程序代理程序以及 Apache ZooKeeper 連接字符串。

使用獲取引導代理 AWS CLI

執行下列命令，取代*ClusterArn*為建立叢集時取得的 Amazon 資源名稱 (ARN)。若您沒有叢集的 ARN，可透過列出所有叢集來找到該 ARN。如需詳細資訊，請參閱 the section called “列出叢集”。

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

針對使用 the section called “IAM 存取控制” 的 MSK 叢集，此命令的輸出如下 JSON 範例所示。

```
{
    "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"
}
```

下列範例顯示已開啟公開存取權限之叢集的引導代理程式。使用`BootstrapBrokerStringPublicSaslIam`進行公用存取，並使用`BootstrapBrokerStringSaslIam`字串從內部進行存取 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"
}
```

引導代理程式字串應包含來自部署 MSK 叢集之可用區域中的三個代理程式 (除非只可使用兩個代理程式)。

使用 API 取得引導代理程式

若要使用 API 取得啟動程式代理程式，請參閱GetBootstrap代理程式。

列出 Amazon MSK 叢集

使用列出叢集 AWS Management Console

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
2. 此表格會顯示此帳戶下目前區域的所有叢集。選擇叢集名稱來檢視其詳細資訊。

使用列出叢集 AWS CLI

執行下列命令。

```
aws kafka list-clusters
```

使用 API 列出叢集

若要使用 API 列出叢集，請參閱ListClusters。

元數據管理

Amazon MSK 支持阿帕奇 ZooKeeper 或卡夫元數據管理模式。

從 Amazon MSK 上的阿帕奇卡夫卡 3.7.x 版，您可以創建使用卡夫模式而不是模式的集群。ZooKeeper KRAFT 型叢集仰賴 Kafka 內的控制器來管理中繼資料。

主題

- ZooKeeper 模式
- 牛皮紙模式

ZooKeeper 模式

Apache ZooKeeper 是一種集中式服務，用於維護配置信息，命名，提供分散式同步以及提供組服務。所有這些類型的服務都以某種形式或其他由分佈式應用程序使用，」包括 Apache Kafka。

如果您的叢集使用 ZooKeeper 模式，您可以使用下列步驟取得 Apache ZooKeeper 連接字串。但是，我們建議您使用連接`BootstrapServerString`到您的集群和 perfom 管理操作，因為該`--zookeeper`標誌已在卡夫卡 2.5 被棄用，並從卡夫卡 3.0 中刪除。

使用取得 Apache ZooKeeper 連線字串 AWS Management Console

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
2. 此表格會顯示此帳戶下目前區域的所有叢集。選擇叢集名稱以檢視其描述。
3. 在叢集摘要頁面上，選擇檢視用戶端資訊。這將顯示啟動程序代理程序以及 Apache ZooKeeper 連接字符串。

使用取得 Apache ZooKeeper 連線字串 AWS CLI

1. 如果您不知道叢集的 Amazon Resource Name (ARN)，可以透過列出帳戶中的所有叢集來找到該 ARN。如需詳細資訊，請參閱 the section called “列出叢集”。
2. 若要取得 Apache ZooKeeper 連線字串以及叢集的其他相關資訊，請執行下列命令，*ClusterArn*以叢集的 ARN 取代。

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

此 `describe-cluster` 命令的輸出如以下 JSON 範例所示。

```
{
    "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"
    }
}
```

上一個 JSON 範例會顯示 `describe-cluster` 命令輸出中的 `ZookeeperConnectString` 金鑰。複製與此金鑰對應的值，並進行儲存以供在叢集上建立主題時使用。

> **Important**
>
> 您的 Amazon MSK 叢集必須`ACTIVE`處於狀態，您才能取得 Apache ZooKeeper 連線字串。當叢集仍處於 `CREATING` 狀態時，`describe-cluster` 命令的輸出不包含 `ZookeeperConnectString`。如果是這種情況，請等待幾分鐘，然後在叢集達到 `ACTIVE` 狀態後再次執行 `describe-cluster`。

使用 API 獲取阿帕奇 ZooKeeper 連接字符串

若要使用 API 取得 Apache ZooKeeper 連接字串，請參閱DescribeCluster。

牛皮紙模式

Amazon MSK 在卡夫卡 3.7.x 版推出了卡夫卡（阿帕奇卡夫卡筏）的支持。阿帕奇卡夫卡社區開發卡夫卡替換 Apache ZooKeeper 的阿帕奇在阿帕奇卡夫卡集群的元數據管理。在 Kraft 模式下，集群元數據是一組卡夫卡控制器，這是卡夫卡集群的一部分，而不是跨節點內傳播。 ZooKeeper 隨附 Kraft 控制器，無需額外付費，無需額外的設定或管理。有關卡夫的更多信息，請參閱 KIP-500。

以下是 MSK 上 Kraft 模式的一些注意事項：

- Kraft 模式僅適用於新叢集。建立叢集後，您就無法切換中繼資料模式。

- 在 MSK 主控台上，您可以選擇 Kafka 版本 3.7.x，然後在叢集建立視窗中選取 Kraft 核取方塊，來建立 KRAFT 型叢集。
- 若要使用 MSK API `CreateCluster`或`CreateClusterV2`作業在 Kraft 模式下建立叢集，您應該使用作`3.7.x.kraft`為版本。用`3.7.x`作在 ZooKeeper 模式下建立叢集的版本。
- 每個代理的分區數量是在 Kraft 和 ZooKeeper 基於集群相同。但是，Kraft 可讓您透過在叢集中佈建更多代理程式，在每個叢集中託管更多分割區。
- 在 Amazon MSK 上使用 Kraft 模式不需要任何 API 變更。但是，如果您的用戶端今天仍然使用`--zookeeper`連接字串，您應該更新用戶端以使用`--bootstrap-server`連接字串來連接到叢集。該`--zookeeper`標誌在阿帕奇卡夫卡 2.5 版本不推薦使用，並從卡夫卡 3.0 版本開始刪除。因此，我們建議您使用最新的 Apache Kafka 用戶端版本和叢集的所有連`--bootstrap-server`接字串。
- ZooKeeper 模式繼續可用於所有發布的版本，其中動物園管理員也支持 Apache 卡夫卡。如支援的 Apache Kafka 版本需停止支援 Apache Kafka 版本和 future 更新的詳細資訊，請參閱。
- 您應該檢查您使用的任何工具是否能夠在沒有連接的情況下 ZooKeeper 使用卡夫卡管理 API。如需將使用 LinkedIn的巡航控制阿帕奇卡夫卡與 Amazon MSK叢集連接至巡航控制的更新步驟，請參閱。巡航控制系統還具有運行巡航控制不帶的說明 ZooKeeper。
- 您無需直接存取叢集的 Kraft 控制器即可執行任何管理動作。不過，如果您使用開放式監控來收集指標，您也需要控制器的 DNS 端點，才能收集叢集的一些與控制器相關的非控制器相關指標。您可以從 MSK 主控台或使用 ListNodesAPI 作業取得這些 DNS 端點。使用 Prometheus 進行開放式監控如需針對 KRAF 型叢集設定開放式監控的更新步驟，請參閱。
- 您不需要額外的CloudWatch 指標來監視透過模式叢集的 Kraft ZooKeeper 模式叢集。MSK 會管理叢集中使用的 Kraft 控制器。
- 您可以使用`--bootstrap-server`連接字串繼續在 Kraft 模式叢集中使用來管理 ACL。您不應該使用`--zookeeper`連接字串來管理 ACL。請參閱Apache Kafka ACL。
- 在 Kraft 模式下，群集的元數據存儲在 Kafka 內的 Kraft 控制器上，而不是外部 ZooKeeper 節點。因此，您不需要像使用節點一樣單獨控制對控制器 ZooKeeper 節點的存取。

儲存體管理

Amazon MSK 提供的功能可協助您在 MSK 叢集上進行儲存管理。

主題

- 分層儲存
- 縱向擴展代理程式儲存空間
- 佈建儲存輸送量

分層儲存

分層儲存是 Amazon MSK 的低成本儲存層級，可擴展到幾乎無限制的儲存空間，讓建置串流資料應用程式具有成本效益。

您可以建立設定了分層儲存的 Amazon MSK 叢集，平衡效能和成本。Amazon MSK 會將串流資料儲存在效能最佳化的主要儲存層中，直到資料達到 Apache Kafka 主題保留期限為止。然後，Amazon MSK 會自動將資料移入新的低成本儲存層。

當您的應用程式開始從分層儲存讀取資料時，您可以預期前幾個位元組的讀取延遲會增加。當您開始從低成本儲存層依序讀取剩餘資料時，您可以預期與主要儲存層近似的延遲。您不需要針對低成本分層儲存佈建任何儲存，也不需要管理基礎設施。您可儲存任意數量的資料，只需按實際使用量付費。此功能與 KIP-405: Kafka Tiered Storage 中介紹的 API 相容。

以下為分層儲存的一些功能：

- 您可以擴展到幾乎無限制的儲存空間。您不必了解如何擴展 Apache Kafka 基礎設施。
- 您可以在 Apache Kafka 主題中將資料保留更長的時間，或增加主題儲存空間，而無需增加代理程式的數量。
- 它提供了更長的持續時間安全緩衝區，以處理過程中的意外延遲。
- 您可以使用現有的串流處理程式碼和 Kafka API，以舊資料的精確生產順序重新處理舊資料。
- 由於次要儲存上的資料不需要跨代理程式磁碟進行複寫，因此分區重新平衡的速度會更快。
- 代理程式與分層儲存之間的資料會在 VPC 內移動，不會透過網際網路傳輸。
- 用戶端機器可以使用相同的程序連線至已啟用分層儲存的新叢集，就像連線到未啟用分層儲存的叢集一樣。請參閱建立用戶端機器。

分層儲存要求

- 您必須使用 Apache Kafka 用戶端 3.0.0 或更高版本，才能建立已啟用分層儲存的新主題。若要將現有主題轉換到分層儲存，您可以重新設定使用 3.0.0 以下版本 (支援的 Apache Kafka 最低版本為 2.8.2.tiered) Kafka 用戶端的用戶端機器，以啟用分層儲存。請參閱步驟 4：建立主題。
- 已啟用分層儲存的 Amazon MSK 叢集必須使用 3.6.0 或更高版本或 2.8.2 版。

分層儲存的限制

分層儲存具有下列限制：

- 分層儲存僅適用於佈建類型叢集。
- 階層式儲存不支援代理程式大小 t3.small。
- 低成本儲存的最短保留期間為 3 天。主要儲存沒有最短保留期間。
- 分層儲存不支援在代理程式上使用多日誌目錄 (JBOD 相關功能)。
- 分層儲存不支援壓縮的主題。確保所有已開啟分層儲存主題都已將 cleanup.policy 設定為「DELETE」。
- 分層儲存可針對個別主題停用，但不能在整個叢集停用。停用後，就無法針對主題重新啟用分層儲存。
- 如果您使用的是 Amazon MSK 2.8.2 版本，則只能遷移到另一個支援階層儲存體的 Apache Kafka 版本。如果您不想繼續使用支援分層儲存的版本，請建立新的 MSK 叢集並將資料移轉至該叢集。
- 此工 kafka-log-dirs 具無法報告階層式儲存資料大小。此工具只會報告主要儲存中的日誌區段大小。

如何將日誌區段複製到分層儲存

為新主題或現有主題啟用分層儲存後，Apache Kafka 會從主要儲存將關閉的日誌區段複製到分層儲存。

- Apache Kafka 僅複製關閉的日誌區段。它會將日誌區段中的所有訊息複製到分層儲存。
- 作用中區段不符合分層的資格。日誌區段大小 (segment.bytes) 或區段滾動時間 (segment.ms) 控制區段關閉的速率，以及 Apache Kafka 之後將它們複製到分層儲存的速率。

啟用分層儲存之主題的保留設定，有別於未啟用分層儲存之主題的設定。下列規則會控制在已啟用分層儲存之主題中訊息的保留：

- 定義 Apache Kafka 中的保留有使用兩種設定：log.retention.ms (時間) 和 log.retention.bytes (大小)。這些設定會決定 Apache Kafka 在叢集中保留的資料總持續時間和大小。無論是否啟用了分層儲存模式，都可以在叢集層級設定這些組態。您可以使用主題組態覆寫主題層級的設定。
- 啟用分層儲存後，您可以另外指定主要高效能儲存層儲存資料的時間長度。例如，如果主題的整體保留 (log.retention.ms) 設定為 7 天而本機保留 (local.retention.ms) 為 12 小時，則叢集主要儲存只會保留資料 12 小時。低成本的儲存層會保留資料滿 7 天。
- 一般保留設定適用於完整日誌。這包含其分層儲存部分和主要儲存部分。
- local.retention.ms 或 local.retention.bytes 設定會控制主要儲存中訊息的保留。資料達到主要儲存對完整日誌的保留設定閾值 (local.retention.ms/bytes) 後，Apache Kafka 會將主要儲存中的資料複製到分層儲存。然後，資料即符合到期資格。

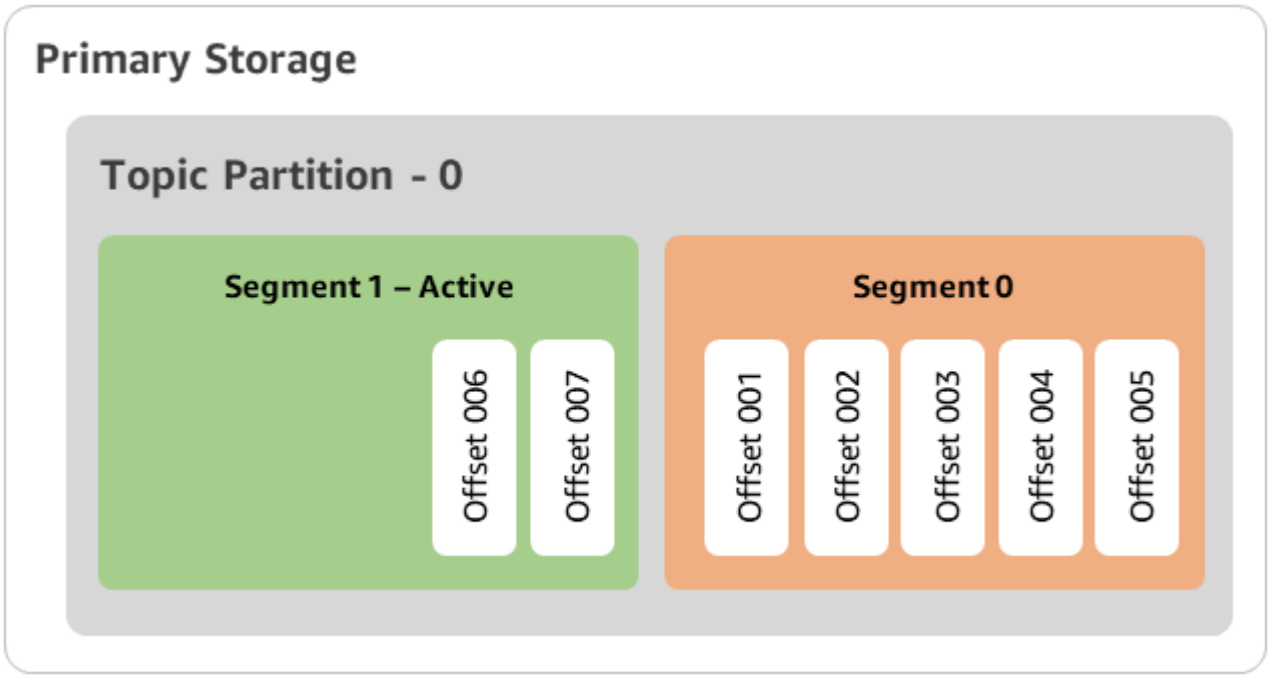
- Apache Kafka 將日誌區段中的訊息複製到分層儲存後，它會根據 retention.ms 或 retention.bytes 設定，從叢集中移除訊息。

分層儲存案例範例

此案例說明啟用分層儲存後，在主要儲存中具有訊息之現有主題的行為方式。將 remote.storage.enable 設定為 `true` 後，可以在此主題上啟用分層儲存。在此範例中，retention.ms 設定為 5 天，local.retention.ms 設定為 2 天。以下是一個區段到期時的事件序列。

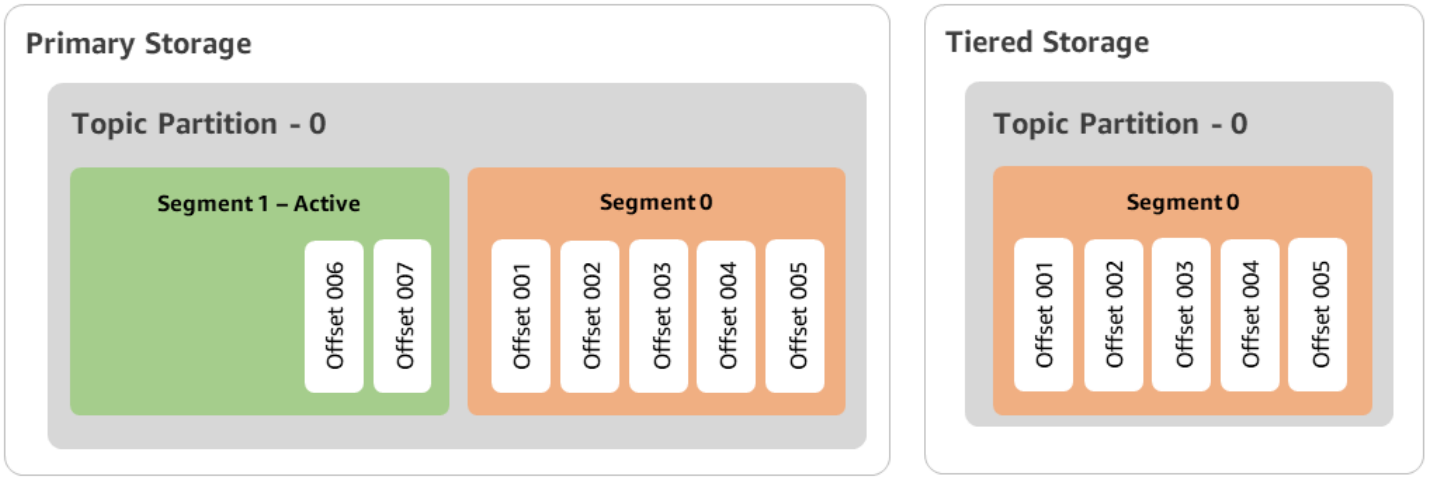
時間 T0 - 啟用分層儲存之前。

在您啟用此主題的分層儲存之前，有兩個日誌區段。其中一個區段在現有主題分區 0 中處於作用中狀態。



時間 T1 (< 2 天) - 已啟用分層儲存。將區段 0 複製到分層儲存。

啟用此主題的分層儲存後，Apache Kafka 會在此區段符合初始保留設定後，將日誌區段 0 複製到分層儲存。Apache Kafka 還會保留區段 0 在主要儲存中的副本。作用中區段 1 尚未符合複製到分層儲存的資格。在此時間表中，Amazon MSK 尚未針對區段 0 和區段 1 中的任何訊息套用任何保留設定。(local.retention.bytes/ms、retention.ms/bytes)



時間 T2 - 本機保留生效。

2 天後，主要保留設定會對 Apache Kafka 複製到分層儲存的區段 0 生效。local.retention.ms 為 2 天的設定決定了這一點。區段 0 現在會在主要儲存中到期。作用中區段 1 既不符合到期資格，也不符合複製到分層儲存的資格。

時間 T3 - 整體保留生效。

5 天後，保留設定生效，Kafka 會從分層儲存中清除日誌區段 0 及關聯的訊息。區段 1 既不符合到期資格，也尚未符合複製到分層儲存的資格，因為區段 1 處於作用中狀態。區段 1 尚未關閉，因此不符合區段滾動的資格。

使用分層儲存建立 Amazon MSK 叢集 AWS Management Console

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
2. 選擇建立叢集。
3. 選擇自訂建立，以便設定分層儲存。
4. 指定叢集的名稱。
5. 在叢集類型中，選取佈建。
6. 選擇一個支援分層儲存的 Amazon Kafka 版本以供 Amazon MSK 用於建立叢集。
7. 指定比卡夫卡 .t3. 小以外的經紀人的大小。
8. 選取您想要 Amazon MSK 在每個可用區域中建立的代理程式數量。下限為每個可用區域 1 個代理程式，上限為每個叢集 30 個代理程式。
9. 指定要分配代理程式的區域數量。
10. 指定每個區域部署的 Apache Kafka 代理程式數量。
11. 選取儲存選項。其中包括分層儲存和 EBS 儲存，以便啟用分層儲存模式。
12. 請依循叢集建立精靈中的其餘步驟進行操作。完成後，分層儲存和 EBS 儲存會在檢閱和建立檢視中顯示為叢集儲存模式。
13. 選取 Create cluster (建立叢集)。

使用分層儲存建立 Amazon MSK 叢集 AWS CLI

若要在叢集上啟用分層儲存，請使用正確的 Apache Kafka 版本和分層儲存的屬性來建立叢集。請依循以下程式碼範例。另外，請完成下一節 建立已啟用分層儲存的 Kafka 主題 中的步驟。

如需有關建立叢集的完整支援屬性清單，請參閱 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"
```

建立已啟用分層儲存的 Kafka 主題

若要完成您在建立已啟用分層儲存的叢集時啟動的程序，請同樣使用稍後程式碼範例中的屬性，建立已啟用分層儲存的主題。專為分層儲存而設的屬性如下：

- `local.retention.ms` (例如，10 分鐘) 適用於時間型保留設定；`local.retention.bytes` 適用於日誌區段大小限制。
- 將 `remote.storage.enable` 設定為 `true` 以啟用分層儲存。

以下組態會使用 local.retention.ms，但是您可以使用 local.retention.bytes 取代此屬性。此屬性會控制 Apache Kafka 將資料從主要儲存複製到分層儲存所需的時間，或是 Apache Kafka 可以複製的位元組數量。如需有關支援組態屬性的詳細資訊，請參閱主題層級組態。

> **Note**
>
> 您必須使用 Apache Kafka 用戶端 3.0.0 及以上版本。這些版本僅在這些 `kafka-topics.sh` 用戶端版本中支援名為 `remote.storage.enable` 的設定。若要為使用舊版 Apache Kafka 的現有主題啟用分層儲存，請參閱章節 啟用現有主題上的分層儲存。

```
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
```

啟用和停用現有主題上的分層儲存

這些章節說明如何為已建立的主題啟用和停用分層儲存。若要建立已啟用分層儲存的新叢集和主題，請參閱使用 AWS Management Console建立具有分層儲存的叢集。

啟用現有主題上的分層儲存

若要啟用現有主題的分層儲存，請使用以下範例中的 `alter` 命令語法。啟用現有主題的分層儲存時，不受限於特定 Apache Kafka 用戶端版本。

```
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'
```

停用現有主題上的分層儲存

若要停用現有主題的分層儲存，請依照啟用分層儲存時的相同順序使用 `alter` 命令語法。

```
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**
>
> 停用分層儲存後，會完全刪除分層儲存中的主題資料。Apache Kafka 會保留主要儲存中的資料，但它仍會根據 `local.retention.ms` 套用主要保留規則。停用主題的分層儲存後，無法再次啟用。如果要停用現有主題的分層儲存，不受限於特定 Apache Kafka 用戶端版本。

使用 CLI 在現有叢集上啟用 AWS 階層式儲存

> **Note**
>
> 您只能在叢集的 log.cleanup.policy 設定為 `delete` 時啟用分層儲存，因為在分層儲存上不支援壓縮主題。您可以稍後將個別主題的 log.cleanup.policy 設定為 `compact` (如果該特定主題還未啟用分層儲存)。如需有關支援組態屬性的詳細資訊，請參閱主題層級組態。

1. 更新 Kafka 版本 - 叢集版本不是簡易整數。若要尋找叢集的目前版本，請使用`DescribeCluster`作業或 `describe-cluster` AWS CLI 指令。範例版本為`KTVPDKIKX0DER`。

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

2. 編輯叢集儲存模式。下列程式碼範例顯示使用 update-storage API，將叢集儲存模式編輯為 TIERED。

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

使用主控台更新現有叢集的分層儲存

> **Note**
>
> 您只能在叢集的 log.cleanup.policy 設定為 delete 時啟用分層儲存，因為在分層儲存上不支援壓縮主題。您可以稍後將個別主題的 log.cleanup.policy 設定為 compact (如果該特定主題還未啟用分層儲存)。如需有關支援組態屬性的詳細資訊，請參閱主題層級組態。

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
2. 前往叢集摘要頁面，然後選擇屬性。
3. 前往儲存區段，然後選擇編輯叢集儲存模式。
4. 選擇分層儲存和 EBS 儲存，然後儲存變更。

縱向擴展代理程式儲存空間

您可以增加每個代理程式的 EBS 儲存體數量。您無法減少儲存空間。

在此向上擴展操作期間，儲存磁碟區仍然可供使用。

> **Important**
>
> 擴展 MSK 叢集的儲存後，會立即提供額外的儲存。但是，在每個儲存擴展事件之後，叢集都需要一段冷卻期間。Amazon MSK 會在此冷卻期間最佳化叢集，然後才能再次擴展叢集。此冷卻期間從最少 6 小時到 24 小時以上不等，具體取決於叢集的儲存大小、使用率以及流量。這適用於使用 S UpdateBrokertorage 操作的 auto 調整規模事件和手動調整規模。如需有關調整儲存大小的資訊，請參閱 最佳實務。

您可以使用分層儲存，將代理程式的儲存縱向擴展至無限量的儲存。請參閱 分層儲存。

主題

自動調整規模

若要自動擴充叢集儲存容量以因應增加的使用量，您可以為 Amazon MSK 設定應用程式自動擴展政策。在自動擴展政策中，您可以設定目標磁碟使用率和最大擴展容量。

在對 Amazon MSK 使用自動擴展之前，應考慮以下事項：

- **Important**

  儲存擴展動作只能每六小時執行一次。

  我們建議您根據自己的儲存需求，從適當調整儲存磁碟區的大小開始。如需有關適當調整叢集大小的指引，請參閱適當調整叢集大小：每個叢集的代理程式數量。
- Amazon MSK 不會為因為使用量的降低而減少叢集儲存容量。Amazon MSK 不支援降低儲存磁碟區的大小。如果您需要降低叢集儲存的大小，則必須將現有叢集遷移至儲存容量較小的叢集。如需有關遷移叢集的相關資訊，請參閱遷移。
- Amazon MSK 不支援在亞太區域 (大阪) 和非洲 (開普敦) 區域使用自動縮減功能。
- 當您將 auto-scaling 政策與叢集建立關聯時，Amazon EC2 Auto Scaling 會自動建立用於目標追蹤的 Amazon CloudWatch 警示。如果您使用 auto-scaling 政策刪除叢集，此 CloudWatch 警示仍會持續存在。若要刪除 CloudWatch 警示，您應該先從叢集移除 auto-scaling 政策，然後再刪除叢集。若要進一步了解目標追蹤，請參閱 Amazon EC2 Auto Scaling User Guide 中的 Target tracking scaling policies for Amazon EC2 Auto Scaling。

自動擴展政策詳細資訊

自動擴展政策會為叢集定義下列參數：

- 儲存使用率目標：Amazon MSK 用來觸發自動擴展操作的儲存使用率閾值。您可以將使用率目標設定為目前儲存容量的 10% 到 80% 之間。建議您將儲存使用率目標設為 50% 到 60% 之間。
- 儲存容量上限：Amazon MSK 可為代理程式儲存容量設定的擴展上限。您可以將每個代理程式的儲存容量上限設為 16 TiB。如需詳細資訊，請參閱 Amazon MSK 配額。

當 Amazon MSK 偵測到 `Maximum Disk Utilization` 指標等於或大於 `Storage Utilization Target` 設定值時，它增加的儲存容量等於兩個數字中的較大者：10 GiB 或目前儲存容量的 10%。例如，如果您有 1000 GiB，則該增加容量為 100 GiB。此服務會每分鐘檢查儲存使用率。進一步擴展操作會持續增加儲存容量，增加容量等於兩個數字中的較大者：10 GiB 或目前儲存容量的 10%。

若要判斷是否已發生 auto-scaling 作業，請使用此 ListClusterOperations作業。

為 Amazon MSK 叢集設定自動擴展

您可以使用 Amazon MSK 主控台、Amazon MSK API，或 AWS CloudFormation 為儲存實作自動擴展。 CloudFormation 支持可通過Application Auto Scaling。

> **Note**
>
> 您無法在建立叢集時實作自動擴展。您必須先建立叢集，然後為叢集建立並啟用自動擴展政策。不過，您可以在 Amazon MSK 服務建立叢集時建立該政策。

使用 AWS Management Console設定自動擴展

1. 登入 AWS Management Console，然後開啟 Amazon MSK 主控台，網址為 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。
2. 在叢集清單中選擇叢集。這會帶您前往列出叢集詳細資訊的頁面。
3. 在針對儲存自動擴展區段，選擇設定。
4. 建立自動擴展政策並為其命名。指定儲存使用率目標、儲存容量上限以及目標指標。
5. 選擇 `Save changes`。

儲存並啟用新政策後，該政策對該叢集會變為作用中狀態。接著，Amazon MSK 會在達到儲存使用率目標時擴充叢集的儲存容量。

使用 CLI 設定自動擴展

1. 使用命 RegisterScalableTarget令註冊儲存區使用率目標。
2. 使用命 PutScalingPolicy令建立自動擴充原則。

使用 API 設定自動擴展

1. 使用 RegisterScalableTargetAPI 註冊儲存使用率目標。

2. 使用 PutScalingPolicyAPI 建立自動擴充原則。

手動擴展

若要增加儲存空間，請等待叢集處於 ACTIVE 狀態。儲存擴展事件之間至少需要間隔六個小時的冷卻期間。即使此操作會立即提供額外的儲存，但服務仍會在叢集上執行最佳化，最長可能需要 24 小時或更長時間。這些最佳化的持續時間與您的儲存大小成正比。

使用擴展代理程式儲存 AWS Management Console

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
2. 選擇您要為其更新代理程式儲存的 MSK 叢集。
3. 在儲存區段中，選擇編輯。
4. 指定您想要的儲存磁碟區。您只能增加儲存空間，無法將其減少。
5. 選擇儲存變更。

使用擴展代理程式儲存 AWS CLI

執行下列命令，取代*ClusterArn*為建立叢集時取得的 Amazon 資源名稱 (ARN)。若您沒有叢集的 ARN，可透過列出所有叢集來找到該 ARN。如需詳細資訊，請參閱 the section called “列出叢集”。

將叢集目前版本取代為 *Current-Cluster-Version*。

> **Important**
>
> 叢集版本不是簡單的整數。若要尋找叢集的目前版本，請使用DescribeCluster作業或描述 AWS CLI 叢集指令。範例版本為KTVPDKIKX0DER。

Target-Volume-in-GiB 參數表示您希望每個代理程式具有的儲存空間大小。只能夠更新所有代理程式的儲存空間。您無法指定個別代理程式來更新其儲存空間。您指定的 *Target-Volume-in-GiB* 的值必須是大於 100 GiB 的整數。更新操作後每個代理程式的儲存空間不能超過 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 縱向擴展代理程式儲存

若要使用 API 更新代理程式儲存體，請參閱UpdateBroker儲存體。

佈建儲存輸送量

Amazon MSK 代理程式會將資料保留在儲存磁碟區上。當生產者寫入叢集、代理程式之間複寫資料，以及在取用者讀取不在記憶體中的資料時，就會取用儲存空間 I/O。磁碟區儲存輸送量是指將資料寫入儲存磁碟區和從儲存磁碟區讀取資料的速率。佈建儲存輸送量功能可在叢集中指定代理程式的速率。

對於代理程式大小或更大的叢集，以及儲存磁碟區是否為 10 GiMiB `kafka.m5.4xlarge` 或更大，您可以指定佈建的輸送量率 (mIB) 每秒。您可以在建立叢集期間指定佈建輸送量。您也可以針對處於 `ACTIVE` 狀態的叢集啟用或停用佈建輸送量。

輸送量瓶頸

代理程式輸送量中的瓶頸有多種原因：磁碟區輸送量、Amazon EC2 到 Amazon EBS 網路輸送量，以及 Amazon EC2 輸出輸送量。您可以啟用佈建儲存輸送量，藉以調整磁碟區輸送量。不過，Amazon EC2 到 Amazon EBS 網路輸送量和 Amazon EC2 輸出輸送量可能導致代理程式輸送量限制。

Amazon EC2 輸出輸送量受到取用者群組數量和每個取用者群組的取用者數量影響。此外，對於較大的代理程式規模，Amazon EC2 到 Amazon EBS 網路輸送量和 Amazon EC2 輸出輸送量都較高。

對於 10 GiB 或更大的磁碟區，您可以佈建每秒 250 MiB 或更高的儲輸送量。預設為每秒 250 MiB。若要佈建儲存體輸送量，您必須選擇代理程式大小 kafka.m5.4xlarge 或更大 (或者 kafka.m7g.2xlarge 或更大)，您可以指定最大輸送量，如下表所示。

經紀人大小	儲存輸送量上限 (MiB/秒)
kafka.m5.4xlarge	593
kafka.m5.8xlarge	850
kafka.m5.12xlarge	1000
kafka.m5.16xlarge	1000
kafka.m5.24xlarge	1000
卡夫卡 .m7 公克 2 倍大	312.5

經紀人大小	儲存輸送量上限 (MiB/秒)
卡夫卡 .m7 克 .4 倍大	625
卡夫卡 .m7 公克 8 倍大	1000
卡夫卡 .m7 公克 12 倍大	1000
卡夫卡 .m7 公克 16 倍大	1000

測量儲存輸送量

您可以使用 VolumeReadBytes 和 VolumeWriteBytes 指標來測量叢集的平均儲存輸送量。這兩個指標的總和就是平均儲存輸送量 (以位元組為單位)。若要取得叢集的平均儲存輸送量，請將這兩個指標設定為 SUM，並將期間設定為 1 分鐘，然後使用下列公式。

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

如需有關 VolumeReadBytes 和 VolumeWriteBytes 指標的資訊，請參閱 the section called “PER_BROKER 層級監控”。

組態更新

您可以在開啟佈建輸送量之前或之後更新 Amazon MSK 組態。但是，在執行以下兩個動作之前，不會看到所需的輸送量：更新 num.replica.fetchers 組態參數並開啟佈建輸送量。

在預設的 Amazon MSK 組態中，num.replica.fetchers 的值為 2。若要更新您的 num.replica.fetchers，可以使用下表中的建議值。這些值僅用於指引。建議您根據使用案例調整這些值。

經紀人大小	num.replica.fetchers
kafka.m5.4xlarge	4
kafka.m5.8xlarge	8
kafka.m5.12xlarge	14

經紀人大小	num.replica.fetchers
kafka.m5.16xlarge	16
kafka.m5.24xlarge	16

更新的組態可能需要 24 小時才能生效，如果來源磁碟區並未充分利用，則可能需要更長的時間。不過，轉換磁碟區效能至少等同於遷移期間來源儲存磁碟區的效能。充分利用的 1 TiB 磁碟區通常需要大約六個小時才能遷移至更新的組態。

使用佈建儲存體輸送量 AWS Management Console

1. 登入 AWS Management Console，然後開啟 Amazon MSK 主控台，網址為 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。
2. 選擇建立叢集。
3. 選擇自訂建立。
4. 指定叢集的名稱。
5. 在儲存區段中，選擇啟用。
6. 選擇每個代理程式的儲存輸送量值。
7. 選擇 VPC、區域、子網路和安全群組。
8. 選擇下一步。
9. 在安全步驟的底部，選擇下一步。
10. 在監控和標籤步驟的底部，選擇下一步。
11. 檢閱叢集設定，然後選擇建立叢集。

使用佈建儲存體輸送量 AWS CLI

本節顯示如何使用建立已啟用佈 AWS CLI 建輸送量的叢集的範例。

1. 複製下方的 JSON 並貼到檔案中。以帳戶中的值取代子網路 ID 和安全群組 ID 預留位置。將檔案命名為 `cluster-creation.json` 並儲存。

```
{
    "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. 從上一個步驟中儲存 JSON 檔案的目錄執行下列 AWS CLI 命令。

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

使用 API 佈建儲存輸送量

若要在建立叢集時設定佈建的儲存體輸送量，請使用 CreateClusterV2。

更新代理大小

您可以根據需求擴展 MSK 叢集，方法是變更代理程式的大小，而無需重新指派 Apache Kafka 分割區。變更代理程式的大小可讓您根據工作負載的變更彈性調整 MSK 叢集的運算容量，而不會中斷叢集 I/O。Amazon MSK 針對指定叢集中的所有代理程式使用相同的代理程式大小。

本節說明如何更新 MSK 叢集的代理程式大小。您可以將叢集代理程式的大小從 M5 或 T3 更新為 M7g，或從 M7g 更新為 M5。請注意，移轉至較小的代理程式大小可能會降低效能並降低每個代理程式可達到的最大輸送量。移轉至較大的代理程式大小可以提高效能，但可能會花費較高。

代理程式大小更新會在叢集啟動並執行時以滾動的方式發生。這表示 Amazon MSK 一次取下一個代理程式來執行代理程式大小更新。如需有關如何在代理程式大小更新期間使叢集高度可用的資訊，請參閱。the section called “建置高可用性叢集”為了進一步減少對生產力的任何潛在影響，您可以在低流量期間執行經紀人規模更新。

在代理程式大小更新期間，您可以繼續產生和使用資料。不過，您必須等到更新完成後，才能重新啟動代理程式或調用 Amazon MSK operations 下列出的任何更新操作。

如果您想要將叢集更新為較小的代理程式大小，建議您先嘗試在測試叢集上進行更新，以查看其對您的案例有何影響。

> **Important**
>
> 如果每個代理程式的分割區數目超過中指定的最大數目，則無法將叢集更新為較小的代理程式大小the section called “ 適當調整叢集大小：每個代理程式的分區數量”。

使用更新代理程式大小 AWS Management Console

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
2. 選擇您要更新代理程式大小的 MSK 叢集。
3. 在叢集的詳細資料頁面上，找到「代理程式摘要」區段，然後選擇 「編輯代理程式大小」。
4. 從列表中選擇所需的經紀人大小。
5. 儲存變更。

使用更新代理程式大小 AWS CLI

1. 執行下列命令，取代*ClusterArn*為建立叢集時取得的 Amazon 資源名稱 (ARN)。若您沒有叢集的 ARN，可透過列出所有叢集來找到該 ARN。如需詳細資訊，請參閱 the section called “列出叢集”。

   將目*##########################TargetType*的新大小取代目前叢集版本。要進一步了解經紀人規模，請參閱the section called “經紀人規模”。

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

   以下是如何使用此命令的範例：

   ```
   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
   ```

   此 命令的輸出如以下 JSON 範例所示。

   ```
   {
       "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. 若要取得update-broker-type作業的結果，請執行下列命令，將 *ClusterOperationArn* 取代為您在命令輸出中取得的 update-broker-type ARN。

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

   此 describe-cluster-operation 命令的輸出如以下 JSON 範例所示。

   ```
   {
     "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` 具有值 `UPDATE_IN_PROGRESS`，請稍候一段時間，然後再次執行 `describe-cluster-operation` 命令。

使用 API 更新代理程式大小

若要使用 API 更新代理程式大小，請參閱UpdateBroker類型。

您可以使用`UpdateBrokerType`將叢集代理程式大小從 M5 或 T3 更新為 M7g，或從 M7g 更新為 M5。

更新 Amazon MSK 叢集的組態

若要更新叢集的配置，請確定叢集處於 `ACTIVE` 狀態。您還必須確保 MSK 叢集上每個代理程式的分區數量低於 the section called “ 適當調整叢集大小：每個代理程式的分區數量” 中所述的限制。您無法更新超過這些限制之叢集的組態。

如需 MSK 組態的資訊，包括如何建立自訂組態、您可以更新哪些屬性，以及更新現有叢集的組態時會發生什麼情況，請參閱 組態。

使用更新叢集的配置 AWS CLI

1. 複製以下 JSON 並將其儲存到檔案。將檔案命名為 `configuration-info.json`。以您要用來更新叢集之組態的 Amazon 資源名稱 (ARN) 取*`ConfigurationArn`*代。ARN 字串必須在下列 JSON 的引號中。

使用您要使用的組態修訂版本取代 *Configuration-Revision*。組態修訂版是從 1 開始的整數 (整數)。這個整數不能在以下 JSON 的引號中。

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

2. 執行下列命令，取代*ClusterArn*為您建立叢集時取得的 ARN。若您沒有叢集的 ARN，可透過列出所有叢集來找到該 ARN。如需詳細資訊，請參閱 the section called “列出叢集”。

   將 *Path-to-Config-Info-File* 取代為組態資訊檔案的路徑。如果您為在上一個步驟中建立的檔案 configuration-info.json 命名，並將其儲存在目前的目錄中，則 *Path-to-Config-Info-File* 會是 configuration-info.json。

   將叢集目前版本取代為 *Current-Cluster-Version*。

   > **Important**
   >
   > 叢集版本不是簡單的整數。若要尋找叢集的目前版本，請使用DescribeCluster作業或描述 AWS CLI 叢集指令。範例版本為KTVPDKIKX0DER。

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

   以下是如何使用此命令的範例：

   ```
   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"
   ```

   此 update-cluster-configuration 命令的輸出如以下 JSON 範例所示。

   ```
   {
       "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. 若要取得update-cluster-configuration作業的結果，請執行下列命令，將 *ClusterOperationArn* 取代為您在命令輸出中取得的 update-cluster-configuration ARN。

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

此 describe-cluster-operation 命令的輸出如以下 JSON 範例所示。

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

在此輸出中，OperationType 是 UPDATE_CLUSTER_CONFIGURATION。如果 OperationState 具有值 UPDATE_IN_PROGRESS，請稍候一段時間，然後再次執行 describe-cluster-operation 命令。

使用 API 更新叢集的組態

若要使用 API 更新叢集的配置，請參閱UpdateCluster組態。

擴充 Amazon MSK 叢集

如果想要增加 MSK 叢集中代理程式的數量，請使用此 Amazon MSK 操作。若要展開叢集，請確定它處於 ACTIVE 狀態。

> **Important**
>
> 如果想要擴充 MSK 叢集，請務必使用此 Amazon MSK 操作。不要嘗試在不使用此操作的情況下將代理程式新增到叢集中。

如需如何在將代理程式新增至叢集後重新平衡分割區的詳細資訊，請參閱 the section called “重新指派分割區”。

使用擴充叢集 AWS Management Console

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
2. 選擇要增加其代理程式數量的 MSK 叢集。
3. 在叢集詳細資訊頁面上，選擇叢集層級代理程式詳細資訊標題旁的編輯按鈕。
4. 輸入想要叢集在每個可用區域擁有的代理程式數量，然後選擇儲存變更。

使用擴充叢集 AWS CLI

1. 執行下列命令，取代*ClusterArn*為建立叢集時取得的 Amazon 資源名稱 (ARN)。若您沒有叢集的 ARN，可透過列出所有叢集來找到該 ARN。如需詳細資訊，請參閱 the section called “列出叢集”。

   將叢集目前版本取代為 *Current-Cluster-Version*。

   > **Important**
   >
   > 叢集版本不是簡單的整數。若要尋找叢集的目前版本，請使用DescribeCluster作業或描述 AWS CLI 叢集指令。範例版本為KTVPDKIKX0DER。

Target-Number-of-Brokers 參數代表您希望叢集在這項操作成功完成時擁有的代理程式節點總數。您指定的 *Target-Number-of-Brokers* 值必須是大於叢集中代理程式目前數量的整數。它也必須是可用區域數量的倍數。

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

此 update-broker-count 操作的輸出如以下 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. 若要取得update-broker-count作業的結果，請執行下列命令，將 *ClusterOperationArn* 取代為您在命令輸出中取得的 update-broker-count ARN。

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

此 describe-cluster-operation 命令的輸出如以下 JSON 範例所示。

```
{
    "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
        }
    }
}
```

在此輸出中，OperationType 是 INCREASE_BROKER_COUNT。如果 OperationState 具有值 UPDATE_IN_PROGRESS，請稍候一段時間，然後再次執行 describe-cluster-operation 命令。

使用 API 擴充叢集

若要使用 API 增加叢集中的代理程式數量，請參閱UpdateBroker計數。

從 Amazon MSK 叢集中移除代理程式

當您想要從 Amazon 阿帕奇卡夫卡受管串流 (MSK) 佈建叢集中移除代理程式時，請使用此 Amazon MSK 作業。您可以移除一組代理程式來減少叢集的儲存和運算容量，而不會影響可用性、資料耐久性風險或資料串流應用程式中斷。

您可以在叢集中新增更多代理程式以處理流量增加，並在流量消退時移除代理程式。透過新增和移除代理程式功能，您可以充分利用叢集容量，並最佳化 MSK 基礎架構成本。移除代理程式可讓您控制現有叢集容量的代理程式層級，以符合工作負載需求，並避免移轉至其他叢集。

使用主 AWS 控台、命令列介面 (CLI)、SDK，或 AWS CloudFormation 減少已佈建叢集的代理程式計數。MSK 會挑選沒有任何分割區的代理程式 (Canary 主題除外)，並防止應用程式向這些代理程式產生資料，同時安全地從叢集中移除這些代理程式。

如果您想要減少叢集的儲存空間和運算能力，則應為每個可用區域移除一個代理程式。例如，您可以在單一代理程式移除作業中，從兩個可用區域叢集移除兩個代理程式，或從三個可用區域叢集中移除三個代理程式。

如需有關如何在從叢集中移除代理程式之後重新平衡分割區的資訊，請參閱the section called “重新指派分割區”。

無論執行個體大小為何，您都可以從所有以 M5 和 M7g 為基礎的 MSK 佈建叢集中移除代理程式。

代理去除在卡夫卡 2.8.1 及以上版本支持，包括在卡夫卡模式集群。

主題

準備通過刪除所有分區刪除經紀人

在開始代理刪除過程之前，首先移動所有分區，除了主題的分區以__amazon_msk_canary及您打算刪除的代理程序__amazon_msk_canary_state中的分區。這些是 Amazon MSK 針對叢集健康狀態和診斷指標所建立的內部主題。

您可以使用卡夫卡管理 API 或巡航控制將分區移動到您打算保留在集群中的其他代理。請參閱重新指派分割區。

移除分割區的範例程序

本節是如何從要刪除的代理程序中刪除分區的示例。假設您有一個包含 6 個代理程式、每個 AZ 中有 2 個代理程式的叢集，並且它有四個主題：

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

1. 建立用戶端電腦，如建立用戶端機器中所述。
2. 設定用戶端機器之後，執行下列命令以列出叢集中所有可用的主題。

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

   在此範例中，我們看到四個主題名稱__amazon_msk_canary__consumer_offsets、__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-c657f7e4ff32-2、、和msk-brk-rmv。
3. 創建一個在客戶端機器topics.json上調用的 json 文件，並添加所有用戶主題名稱，如下面的代碼示例。您不需要包含__amazon_msk_canary主題名稱，因為這是服務管理主題，必要時會自動移動。

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

4. 運行以下命令以生成將分區移動到集群上 6 個代理程序中僅 3 個代理商的建議。

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

5. 創建一個名為的文件reassignment-file.json並複製proposed partition reassignment configuration你從上面的命令。
6. 執行下列命令以移動您在中指定的分割區reassignment-file.json。

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

輸出結果類似如下：

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

7. 運行以下命令以驗證所有分區都已移動。

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

輸出結果類似如下。監視狀態，直到您要求的主題中的所有分割區都成功重新指派為止：

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

8. 當狀態指出每個分割區的分割區重新指派已完成時，請監視 5 分鐘的UserPartitionExists測量結果，以確保它會顯示0給您移動分割區的代理程式。確認完成後，您可以繼續從叢集中移除 Broker。

使用管理主控台移除代 AWS 理程式

使用管理主控台移除代 AWS 理程式

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
2. 選擇包含您要移除之代理程式的 MSK 叢集。
3. 在叢集詳細資料頁面上，選擇動作按鈕，然後選取編輯代理程式數目選項。
4. 輸入您希望叢集擁有每個可用區域的代理程式數目。主控台會摘要列出將要移除的可用區域中代理程式數目。確保這是你想要的。
5. 選擇儲存變更。

為了防止意外刪除代理程序，控制台要求您確認要刪除代理程序。

使用 AWS CLI 移除代理程式

執行下列命令，取代ClusterArn為建立叢集時取得的 Amazon 資源名稱 (ARN)。若您沒有叢集的 ARN，可透過列出所有叢集來找到該 ARN。如需詳細資訊，請列出 Amazon MSK 叢集。Current-Cluster-Version以目前版本的叢集取代。

> **Important**
>
> 叢集版本不是簡單的整數。若要尋找叢集的目前版本，請使用DescribeCluster作業或描述 AWS CLI 叢集指令。範例版本為KTVPDKIKX0DER。

Target-Number-of-Brokers 參數代表您希望叢集在這項操作成功完成時擁有的代理程式節點總數。您為目標數######指定的值必須是小於叢集中目前代理程式數目的整數。它也必須是可用區域數量的倍數。

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

此 update-broker-count 操作的輸出如以下 JSON 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": "DECREASE_BROKER_COUNT",
        "SourceClusterInfo": {
"NumberOfBrokerNodes": 12
        },
        "TargetClusterInfo": {
"NumberOfBrokerNodes": 9
        }
    }
}
```

在此輸出中，OperationType 是 DECREASE_BROKER_COUNT。如果 OperationState 具有值 UPDATE_IN_PROGRESS，請稍候一段時間，然後再次執行 describe-cluster-operation 命令。

使用 AWS API 移除代理程式

若要使用 API 移除叢集中的代理程式，請參閱 Amazon Apache 卡夫卡 API 受管串流參考中的UpdateBroker計數。

更新叢集的安全設定

使用此 Amazon MSK 操作來更新 MSK 叢集的身分驗證與用戶端代理程式加密設定。您也可以更新用來簽署雙向 TLS 身分驗證憑證的私有安全性授權機構。您無法變更叢集內 (代理程式對代理程式) 加密設定。

叢集必須處於 ACTIVE 狀態，才能更新其安全設定。

如果您開啟使用 IAM、SASL 或 TLS 進行身分驗證，您也必須開啟用戶端和代理程式之間的加密。下表顯示可能的組合。

身分驗證	用戶端-代理程式加密選項	代理程式-代理程式加密
Unauthenticated	TLS、PLAINTEXT、TLS_PLAINTEXT	可以開啟或關閉。
mTLS	TLS、TLS_PLAINTEXT	必須為開啟。
SASL/SCRAM	TLS	必須為開啟。
SASL/IAM	TLS	必須為開啟。

如果用戶端-代理程式加密設定為 TLS_PLAINTEXT 且用戶端-身分驗證設定為 mTLS，Amazon MSK 會建立兩種類型的接聽程式以供用戶端連線：一個接聽程式供用戶端使用帶 TLS 加密的 mTLS 身分驗證進行連線，另一個接聽程式供用戶端在不使用身分驗證或加密 (純文字) 的情況下進行連線。

如需有關安全設定的詳細資訊，請參閱 安全。

使用更新叢集的安全性設定 AWS Management Console

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
2. 選擇您要更新的 MSK 叢集。
3. 在安全設定區段中，選擇編輯。
4. 選擇您要用於叢集的身分驗證和加密設定，然後選擇儲存變更。

使用更新叢集的安全性設定 AWS CLI

1. 建立 JSON 檔案，其中包含您想要叢集擁有的加密設定。以下是範例。

   > **Note**
   >
   > 您只能更新用戶端-代理程式加密設定。無法更新叢集內 (代理程式對代理程式) 加密設定。

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

2. 建立 JSON 檔案，其中包含您想要叢集擁有的身分驗證設定。以下是範例。

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

3. 執行下列 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
```

此 update-security 操作的輸出如以下 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. 若要查看update-security作業的狀態，請執行下列命令，將 *ClusterOperationArn* 取代為您在命令輸出中取得的 update-security ARN。

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

此 describe-cluster-operation 命令的輸出如以下 JSON 範例所示。

```
{
    "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 具有值 PENDING 或 UPDATE_IN_PROGRESS，請稍候一段時間，然後再次執行 describe-cluster-operation 命令。

使用 API 更新叢集的安全設定

若要使用 API 更新叢集的安全性設定，請參閱UpdateSecurity。

> **Note**
>
> 用於更新叢集安全性設定的 AWS CLI 和 API 作業是冪等的。這表示，如果您調用安全更新操作，並指定與叢集目前擁有之設定相同的身分驗證或加密設定，則該設定不會變更。

重新啟動 Amazon MSK 叢集的代理程式

當您想要重新啟動 MSK 叢集的代理程式時，請使用此 Amazon MSK 操作。如要重新啟動叢集的代理程式，請確認叢集處於 ACTIVE 狀態。

Amazon MSK 服務可能會在系統維護期間 (例如修補或版本升級) 重新啟動 MSK 叢集的代理程式。手動重新啟動代理程式可讓您測試 Kafka 用戶端的復原能力，以判斷其回應系統維護的方式。

使用重新啟動代理 AWS Management Console

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
2. 選擇您要重新啟動其代理程式的 MSK 叢集。
3. 向下捲動至代理程式詳細資訊區段，然後選擇要重新啟動的代理程式。
4. 選擇重新啟動代理程式按鈕。

使用重新啟動代理 AWS CLI

1. 執行下列命令，取代*ClusterArn*為您建立叢集時取得的 Amazon 資源名稱 (ARN)，以及要重新開機之代理程式的 ID 取代。*BrokerId*

> **Note**
>
> 此 `reboot-broker` 操作一次僅支援重新啟動一個代理程式。

若您沒有叢集的 ARN，可透過列出所有叢集來找到該 ARN。如需詳細資訊，請參閱 the section called “列出叢集”。

若您沒有叢集的代理程式 ID，可透過列出代理程式節點來找到它們。如需詳細資訊，請參閱 list-nodes。

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

此 `reboot-broker` 操作的輸出如以下 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. 若要取得`reboot-broker`作業的結果，請執行下列命令，並*`ClusterOperationArn`*以您在命`reboot-broker`令輸出中取得的 ARN 取代。

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

此 `describe-cluster-operation` 命令的輸出如以下 JSON 範例所示。

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

完成重新啟動操作後，OperationState 為 REBOOT_COMPLETE。

使用 API 重新啟動代理程式

若要使用 API 重新啟動叢集中的代理程式，請參閱RebootBroker。

在修補和其他維護期間，代理程式重新啟動

Amazon MSK 會定期更新代理程式上的軟體。如果您遵循最佳做法，這些更新不會影響應用程式的寫入和讀取。

Amazon MSK 針對軟體使用滾動式更新來維持叢集的高可用性。在此過程中，經紀人一次重新啟動一個，卡夫卡會自動將領導層轉移到另一個在線經紀人。Kafka 客戶端具有內置機制，可以自動檢測分區領導層的變化，並繼續將數據寫入 MSK 集群。

經紀人離線後，在客戶端上看到暫時斷開連接錯誤是正常的。您還將觀察到一個簡短的窗口（最多 2 分鐘，通常更少）p99 讀取和寫入延遲（通常是高毫秒，最多約 2 秒）的一些尖峰。這些峰值是預期的，並且是由客戶重新連接到新的領導者代理引起的。它不會影響您的產品或消費，並且會在重新連接後解決問題。

您也會發現測量結果有所增加UnderReplicatedPartitions，因為已關閉的中介程式上的分割區不再複寫資料，這是預期的。這對應用程序的寫入和讀取作為這些分區託管在其他代理程序現在正在服務請求的這些分區的複本沒有影響。

軟件更新後，當代理恢復聯機時，它需要「catch」離線時產生的消息。在 catch 期間，您也可能會觀察到磁碟區輸送量和 CPU 的使用率有所增加。如果您的代理程式上有足夠的 CPU、記憶體、網路和磁碟區資源，這些資源應該不會影響叢集的寫入和讀取。

標記 Amazon MSK 叢集

您可以使用標籤的形式，將自己的中繼資料指派至 Amazon MSK 資源，例如 MSK 叢集。標籤是您為資源所定義的索引鍵值組。使用標籤是管理 AWS 資源和組織資料 (包括帳單資料) 的簡單而強大的方式。

主題

標籤基本概念

您可使用 Amazon MSK API 完成下列任務：

- 將標籤新增至 Amazon MSK 資源。
- 列出 Amazon MSK 資源的標籤。
- 從 Amazon MSK 資源移除標籤。

您可以使用標籤來分類 Amazon MSK 資源。例如，您可以依用途、擁有者或環境來分類 Amazon MSK 叢集。由於您定義了每個標籤的金鑰和值，您可以建立一組自訂的類別，以符合您的特定需求。例如，您可以定義一組標籤，協助您根據擁有者和關聯的應用程式來追蹤叢集。

下列為數個標籤的範例：

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

使用標記追蹤成本

您可以使用標籤來分類和追蹤 AWS 成本。將標籤套用至包括 Amazon MSK 叢集在內的 AWS 資源時，您的 AWS 成本分配報告會包含按標籤彙總的用量和成本。套用代表商業類別的標籤 (例如成本中

心、應用程式名稱或擁有者)，即可整理多個服務的成本。如需詳細資訊，請參閱《AWS Billing 使用者指南》中的將成本分配標籤用於自訂帳單報告。

標籤限制

下列限制適用於 Amazon MSK 內的標籤。

基本限制

- 每一資源標籤數最多為 50。
- 標籤金鑰與值皆區分大小寫。
- 您無法變更或編輯已刪除資源的標籤。

標籤鍵限制

- 每個標籤鍵都必須是唯一的。如果您新增具有已使用金鑰的標籤，則新的標籤會覆寫現有金鑰值對。
- 標籤金鑰開頭不能為 `aws:`，因為此字首保留供 AWS使用。 AWS 會代表您建立開頭為此字首的標籤，但您無法加以編輯或刪除。
- 標籤鍵的長度必須介於 1 到 128 個 Unicode 字元之間。
- 標籤鍵必須包含下列字元：Unicode 字母、數字、空格以及下列特殊字元：`_ . / = + - @`。

標籤值限制

- 標籤值的長度必須介於 0 到 255 個 Unicode 字元之間。
- 標籤值可以空白。否則，它們必須包含下列字元：Unicode 字母、數字、空格以及下列任何特殊字元：`_ . / = + - @`。

使用 Amazon MSK API 標記資源

您可以使用下列操作來標記或取消標記 Amazon MSK 資源，或列出資源目前的標籤組：

- ListTagsForResource
- TagResource
- UntagResource

Amazon MSK 組態

適用於 Apache Kafka 的 Amazon 受管串流可為代理程式、主題和 Apache ZooKeeper 節點提供預設組態。您也可建立自訂組態，將其用於建立新的 MSK 叢集或更新現有叢集。MSK 組態包含一組屬性及其對應值。

主題

自訂 MSK 組態

Amazon MSK 可讓您建立自訂 MSK 組態，並在其中設定下列屬性。未明確設定其值的屬性將採用 the section called “預設組態” 內的值。如需組態屬性的詳細資訊，請參閱 Apache Kafka 組態。

Apache Kafka 組態屬性

名稱	描述
allow.everyone.if.no.acl.found	如果要將此屬性設定為 `false`，請先確定您已為叢集定義 Apache Kafka ACL。如果將此屬性設定為 `false`，但卻並未先定義 Apache Kafka ACL，您就會失去對叢集的存取權。如果發生這種狀況，您可以再次更新組態並將此屬性設定為 `true`，以重新獲得對叢集的存取權。
auto.create.topics.enable	啟用伺服器的主題自動建立功能。
compression.type	特定主題的最終壓縮類型。您可將此屬性設定為標準壓縮轉碼器 (`gzip`、`snappy`、`lz4` 和 `zstd`)。它還可接受 `uncompressed` 。這個值相當於無壓縮。如果將該值設定為 `producer`，就代表保留生產者設定的原始壓縮轉碼器。

名稱	描述
connections.max.idle.ms	閒置連線逾時 (以毫秒計)。伺服器插槽處理器執行緒會關閉閒置時間超過此屬性設定值的連線。
default.replication.factor	自動建立主題的預設複寫係數。
delete.topic.enable	啟用刪除主題操作。若此設定關閉，您將無法透過管理員工具刪除主題。
group.initial.rebalance.delay.ms	群組協調員在首次執行重新平衡前等待更多資料取用者加入新群組的時間。延遲越長，意味著重新平衡的次數越少，但處理開始前的等待時間會增加。
group.max.session.timeout.ms	已註冊取用者工作階段逾時值上限。逾時值越大，讓取用者處理活動訊號間的訊息的時間越長，偵測故障耗費的時間也越長。
group.min.session.timeout.ms	已註冊取用者工作階段逾時值下限。逾時值越小，故障偵測的速度越快，但取用者活動訊號會越頻繁。這可能會耗盡代理程式的資源。
leader.imbalance.per.broker.percentage	每個代理程式允許的領導者不平衡比率。如果每個代理程式的控制器超過此值，控制器會觸發領導者不平衡。此值是以百分比指定。
log.cleaner.delete.retention.ms	您希望 Apache Kafka 保留刪除記錄的時間。最小值為 0。
log.cleaner.min.cleanable.ratio	此組態屬性的值介於 0 和 1 之間。此值決定日誌壓縮器嘗試清除日誌的頻率 (如果已啟用日誌壓縮)。根據預設，如果超過 50% 的日誌已壓縮，Apache Kafka 會避免清除日誌。這個比率限制了日誌重複浪費的空間上限 (如果是 50%，意味著最多 50% 的日誌可能是重複的)。比率越高，意味著清除作業次數越少但清除效率越高，但日誌中浪費的空間也越多。

名稱	描述
log.cleanup.policy	保留時段外的區段預設清除政策。以逗號分隔的有效政策清單。有效政策為 `delete` 和 `compact`。對於已啟用分層儲存的叢集，有效政策僅限 `delete`。
log.flush.interval.messages	訊息排清到磁碟前，日誌分區上累積的訊息數量。
log.flush.interval.ms	訊息排清到磁碟前，主題訊息保留在記憶體內的時間上限 (毫秒)。如未設定此值，將採用 log.flush.scheduler.interval.ms 的值。最小值為 0。
log.message.timestamp.difference.max.ms	代理程式接收訊息的時間戳記與訊息中指定的時間戳記之時間差值上限。如果 log.message. 時間戳記類型 =CreateTime，如果時間戳記的差異超過此閾值，則會拒絕訊息。如果記錄. 消息. 時間戳類型 = 時間，則忽略此配置。LogAppend
log.message.timestamp.type	指定訊息內的時間戳記是訊息建立時間還是日誌附加時間。允許的值為 `CreateTime` 和 `LogAppendTime` 。
log.retention.bytes	日誌被刪除前的大小上限。
log.retention.hours	日誌檔案被刪除前的保留時數，為 log.retention.ms 屬性的第三順位值。
log.retention.minutes	日誌檔案被刪除前的保留分鐘數，為 log.retention.ms 屬性的第二順位值。如未設定此值，將採用 log.retention.hours 的值。
log.retention.ms	日誌檔案被刪除前的保留時間 (毫秒)，如未設定，將採用 log.retention.minutes 的值。

名稱	描述
log.roll.ms	新日誌區段推出前的最長時間 (毫秒)。如未設定此屬性，將採用 log.roll.hours 的值。此屬性的最小可能值為 1。
log.segment.bytes	單一日誌檔案的大小上限。
max.incremental.fetch.session.cache.slots	所保持的增量擷取工作階段數量上限。
message.max.bytes	Kafka 允許的最大記錄批次大小。若此值增加，且有取用者的版本低於 0.10.2，則必須上調取用者的擷取大小，取用者才能擷取此大小的記錄批次。 在最新的訊息格式版本中，訊息一律會分組進不同批次，以增進效率。在之前的訊息格式版本中，未壓縮的記錄不會分組進批次，當時，此限制僅適用於單一記錄。 可使用主題層級 max.message.bytes 組態，為每個主題設定此值。
min.insync.replicas	生產者將 ACK 設定為 "all" (或 "-1") 時，min.insync.replicas 的值會指定寫入視為成功前，須接受的最小複本數量。如果無法達到此最小值，則生產者會引發異常（ NotEnoughReplicas 或 NotEnoughReplicasAfterAppend ）。 您可以利用 min.insync.replicas 和 ACK 當中的值強制執行更大數量的耐久性保證。例如，您可以建立複寫係數為 3 的主題，將 min.insync.replicas 設定為 2，ACK 設為 "all" 來產生。若多數複本未接收寫入，如此可確保生產者會引發例外狀況。
num.io.threads	伺服器用於處理要求的執行緒數量，可能包含磁碟 I/O。

名稱	描述
num.network.threads	伺服器從網路接收請求並對其傳送回應所用的執行緒數量。
num.partitions	每個主題的日誌磁碟分割預設數量。
num.recovery.threads.per.data.dir	啟動時日誌復原和關機時排清日誌所用的每個資料目錄執行緒數量。
num.replica.fetchers	從來源代理程式複寫訊息所用的擷取執行緒數量。此值若增加，追隨代理程式的 I/O 平行處理程度會增加。
offsets.retention.minutes	取用者群組遺失所有取用者 (即變為空白) 後，其偏移會保留此值指定的時間，之後才會被捨棄。對獨立 (即手動指派) 取用者而言，偏移會在最後一次遞交加上此保留期間後過期。
offsets.topic.replication.factor	偏移主題的複寫係數。將此值設定得更高能確保可用性。在叢集大小符合此複寫係數要求前，內部主題建立會失敗。
replica.fetch.max.bytes	每個分割區嘗試擷取之訊息位元組數。此非絕對數量上限。若擷取的第一個非空白分區的第一個記錄批次大於此值，則會傳回此記錄批次，確保進度。message.max.bytes (代理程式組態) 或 max.message.bytes (主題組態) 會定義代理程式接受的記錄批次大小上限。
replica.fetch.response.max.bytes	整個擷取回應預期的位元組數量上限。記錄會分批次擷取，若擷取的第一個非空白分區的第一個記錄批次大於此值，則會傳回此記錄批次，確保進度。此非絕對數量上限。message.max.bytes (代理程式組態) 或 max.message.bytes (主題組態) 屬性會指定代理程式接受的最大記錄批次大小。

名稱	描述
replica.lag.time.max.ms	如果追隨者並未傳送任何擷取請求，或是並未在至少這個毫秒數，使用到領導者的日誌端偏移上限，則領導者會從 ISR 中移除跟隨者。 MinValue: 10000 MaxValue = 30000
replica.selector.class	實作 ReplicaSelector的完整類別名稱。代理程式會使用此值尋找偏好的讀取複本。如果您使用 Apache Kafka 2.4.1 及以上版本，且想要允許取用者從最接近的複本進行擷取，請將此屬性設為 `org.apache.kafka.common.replica.RackAwareReplicaSelector` 。如需詳細資訊，請參閱 the section called “Apache Kafka 2.4.1 版 (改為使用 2.4.1.1 版)”。
replica.socket.receive.buffer.bytes	網路請求的插槽接收緩衝。
socket.receive.buffer.bytes	通訊端伺服器通訊埠的 SO_RCVBUF 緩衝區。您可以為此屬性設定的最小值為 -1。如果值為 -1，則 Amazon MSK 會使用作業系統預設值。
socket.request.max.bytes	通訊端要求內的位元組數量上限。
socket.send.buffer.bytes	通訊端伺服器通訊埠的 SO_SNDBUF 緩衝區。您可以為此屬性設定的最小值為 -1。如果值為 -1，則 Amazon MSK 會使用作業系統預設值。
transaction.max.timeout.ms	交易的最大逾時值。如果用戶端要求的交易時間超過此值，代理程式會在中傳回錯誤 InitProducerIdRequest。如此可避免用戶端逾時時間過長，拖延取用者讀取交易中的主題。
transaction.state.log.min.isr	覆寫交易主題的 min.insync.replicas 組態。

名稱	描述
transaction.state.log.replication.factor	交易主題的複寫係數。設定此屬性為更高的值，能增加可用性。在叢集大小符合此複寫係數要求前，內部主題建立會失敗。
transactional.id.expiration.ms	在交易協調器將交易 ID 視為過期之前，等待接收目前交易的交易狀態更新所用的時間 (毫秒)。此設定也會影響生產者 ID 過期時間，因為只要在指定生產者 ID 上次寫入後過了這麼多時間，該生產者 ID 就會過期。如果生產者 ID 的上次寫入由於主題的保留設定而被刪除，該生產者 ID 可能會提早過期。此屬性的最小值為 1 毫秒。
unclean.leader.election.enable	指示不在 ISR 集中的複本是否應該作為領導者與最後手段，即使這可能會導致資料遺失。
zookeeper.connection.timeout.ms	ZooKeeper 模式集群。用戶端等待建立連線的時間上限。 ZooKeeper如未設定此值，將採用 zookeeper.session.timeout.ms 的值。 MinValue = 6000 MaxValue （包括在內）= 一萬八千
zookeeper.session.timeout.ms	ZooKeeper 模式集群。Apache ZooKeeper 工作階段逾時 (以毫秒計)。 MinValue = 6000 MaxValue （包括在內）= 一萬八千

若要了解如何建立自訂 MSK 組態、列出所有組態或描述它們，請參閱 the section called “組態操作”。若要使用自訂 MSK 組態建立 MSK 叢集，或使用新的自訂組態更新叢集，請參閱運作方式。

使用自訂 MSK 組態更新現有 MSK 叢集時，Amazon MSK 會在必要時以滾動方式重新啟動，並依最佳實務最小化客戶停機時間。例如，Amazon MSK 重新啟動每個代理程式後，會試圖讓每個代理程式補上組態更新期間錯過的資料，之後才會移往下一個代理程式。

動態組態

除了 Amazon MSK 提供的組態屬性，您亦可動態設定無須重新啟動代理程式的叢集層級和代理程式層級的組態屬性。您可以動態設定某些組態屬性。這些是在 Apache Kafka 文件內代理程式組態下表格中未標記為唯讀的組態屬性。如需有關動態組態及範例命令的資訊，請參閱 Apache Kafka 文件中的 Updating Broker Configs。

> **Note**
>
> 您可以設定 `advertised.listeners` 屬性，但不能設定 `listeners` 屬性。

主題層級組態

您可使用 Apache Kafka 命令來設定或修改新的和現有主題的主題層級組態屬性。如需有關主題層級組態屬性的詳細資訊及其設定範例，請參閱 Apache Kafka 文件中的 Topic-Level Configs。

組態狀態

Amazon MSK 組態可以是下列狀態之一。若要對組態執行操作，組態必須處於 `ACTIVE` 或 `DELETE_FAILED` 狀態：

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

Amazon MSK 預設組態

建立 MSK 叢集且未指定自訂 MSK 組態時，Amazon MSK 會建立並使用預設組態，組態中的值則列於下表中。對於不在此表格中的屬性，Amazon MSK 會使用與您的 Apache Kafka 版本相關聯的預設值。如需這些預設值的清單，請參閱 Apache Kafka Configuration。

預設組態值

名稱	描述	非分層儲存叢集的預設值	已啟用分層儲存之叢集的預設值
allow.everyone.if.no.acl.found	如果沒有資源模式符合特定資源，資源就	`true`	`true`

名稱	描述	非分層儲存叢集的預設值	已啟用分層儲存之叢集的預設值
	沒有相關聯的 ACL。在此情況下，若此屬性設定為 `true`，則所有使用者均可存取資源，而非僅限超級使用者。		
auto.create.topics.enable	在伺服器上啟用主題自動建立功能。	`false`	`false`
auto.leader.rebalance.enable	啟用自動領導者平衡。背景執行緒會視需要以固定間隔檢查並啟動領導者平衡。	`true`	`true`
default.replication.factor	自動建立主題的預設複寫係數。	如果是 3 個可用區域中的叢集，則是 3，如果是 2 個可用區域中的叢集，則是 2。	如果是 3 個可用區域中的叢集，則是 3，如果是 2 個可用區域中的叢集，則是 2。

名稱	描述	非分層儲存叢集的預設值	已啟用分層儲存之叢集的預設值
local.retention.bytes	刪除舊區段之前分區的本機日誌區段大小上限。如未設定此值，將採用 log.retention.bytes 的值。有效值應該一律小於或等於 log.retention.bytes 值。預設值 -2 指示本機保留沒有限制。這對應於 -1 的 retention.ms/bytes 設定值。local.retention.ms 和 local.retention.bytes 的屬性與 log.retention 類似，因為它們都是用於確定日誌區段應該在本地儲存保留多長時間。現有的 log.retention.* 組態是主題分區的保留組態。這同時包括本地和遠程儲存。有效值：[-2; +Inf] 中的整數	-2 表示無限制	-2 表示無限制

名稱	描述	非分層儲存叢集的預設值	已啟用分層儲存之叢集的預設值
local.retention.ms	本機日誌區段刪除前保留的時間 (毫秒)。如未設定此值，Amazon MSK 將採用 log.retention.ms 的值。有效值應該一律小於或等於 log.retention.bytes 值。預設值 -2 指示本機保留沒有限制。這對應於 -1 的 retention.ms/bytes 設定值。local.retention.ms 和 local.retention.bytes 的值類似 log.retention。MSK 會使用此組態來確定日誌區段應該在本地儲存保留多長時間。現有的 log.retention.* 組態是主題分區的保留組態。這同時包括本地和遠程儲存。有效值為大於 0 的整數。	-2 表示無限制	-2 表示無限制

名稱	描述	非分層儲存叢集的預設值	已啟用分層儲存之叢集的預設值
log.message.timestamp.difference.max.ms	代理程式接收訊息的時間戳記與訊息中指定的時間戳記之最大允許差值。如果 log.message. 時間戳記類型 =CreateTime，則如果時間戳記中的差異超過此閾值，則會拒絕消息。如果記錄. 消息. 時間戳類型 = 時間，則忽略此配置。LogAppend允許的時間戳記差值上限不應該大於 log.retention.ms，以避免不必要地頻繁滾動日誌。	9223372036854775807	對於 Kafka 2.8.2.tiered，應是 86400000
log.segment.bytes	單一日誌檔案的大小上限。	1073741824	134217728

名稱	描述	非分層儲存叢集的預設值	已啟用分層儲存之叢集的預設值
min.insync.replicas	生產者將 ACK (生產者從 Kafka 代理程式獲取的確認) 值設定為 "all" (或 "-1") 後，min.insync.replicas 的值會指定確認寫入視為成功前，須接受的複本數量下限。如果此值不符合此最小值，則生產者會引發例外狀況 (NotEnoughReplicas 或 NotEnough ReplicasAfterAppend)。 若同時使用 min.insync.replicas 和 ACK 的值，您可強制執行更大數量的耐用性保證。例如，您可以建立複寫係數為 3 的主題，將 min.insync.replicas 設定為 2，ACK 設為 "all" 來產生。若多數複本未接收寫入，如此可確保生產者會引發例外狀況。	如果是 3 個可用區域中的叢集，則是 2，如果是 2 個可用區域中的叢集，則是 1。	如果是 3 個可用區域中的叢集，則是 2，如果是 2 個可用區域中的叢集，則是 1。
num.io.threads	伺服器用於生產請求的執行緒數量，可能包含磁碟 I/O。	8	最大值 (8 個，vCPU)，vCPU 數量取決於代理程式的執行個體大小

名稱	描述	非分層儲存叢集的預設值	已啟用分層儲存之叢集的預設值
num.network.threads	伺服器用於從網路接收請求並對網路傳送回應所用的執行緒數量。	5	最大值 (5 個，vCPU/2 個)，vCPU 數量取決於代理程式的執行個體大小
num.partitions	每個主題的日誌磁碟分割預設數量。	1	1
num.replica.fetchers	用於從來源代理程式複寫訊息的擷取器執行緒數量。增加此值就可以增加追隨代理程式中的 I/O 平行處理程度。	2	最大值 (2 個，vCPU/4 個)，vCPU 數量取決於代理程式的執行個體大小
remote.log.msk.disable.policy	與 remote.storage.enable 搭配使用以停用分層儲存。將此政策設定為「刪除」即指示，當您將 remote.storage.enable 設定為 false 時，會刪除分層儲存中的資料。	N/A	DELETE
remote.log.reader.threads	遠端日誌讀取器執行緒集區大小，用於排程從遠端儲存擷取資料的任務。	N/A	最大值 (10 個，vCPU * 0.67)，vCPU 數量取決於代理程式的執行個體大小

名稱	描述	非分層儲存叢集的預設值	已啟用分層儲存之叢集的預設值
remote.storage.enable	如果設定為 true，會啟用主題的分層 (遠端) 儲存。如果設定為 false 且 remote.log.msk.disable.policy 設定為「刪除」，會停用主題層級分層儲存。停用分層儲存時，會從遠端儲存刪除資料。主題的分層儲存停用後就無法再次啟用。	false	true
replica.lag.time.max.ms	如果追隨者並未傳送任何擷取請求，或是並未在至少這個毫秒數，使用到領導者的日誌端偏移上限，則領導者會從 ISR 中移除跟隨者。	30000	30000

名稱	描述	非分層儲存叢集的預設值	已啟用分層儲存之叢集的預設值
retention.ms	必要欄位。最短時間為 3 天。沒有預設值，因為該設定是強制性的。 Amazon MSK 使用保留 retention.ms 的值搭配 local.retention.ms 來確定資料從本機移至分層儲存的時間。local.retention.ms 的值指定何時將資料從本地移至分層儲存。retention.ms 的值指定何時從分層儲存移除資料 (也就是從叢集中移除)。有效值：[-1; +Inf] 中的整數	至少 259,200,000 毫秒 (3 天)。-1 表示無限保留。	至少 259,200,000 毫秒 (3 天)。-1 表示無限保留。
socket.receive.buffer.bytes	通訊端伺服器通訊埠的 SO_RCVBUF 緩衝區。若此值為 -1，將採用 OS 預設值。	102400	102400
socket.request.max.bytes	通訊端要求內的位元組數量上限。	104857600	104857600
socket.send.buffer.bytes	通訊端伺服器通訊埠的 SO_SNDBUF 緩衝區。若此值為 -1，將採用 OS 預設值。	102400	102400

名稱	描述	非分層儲存叢集的預設值	已啟用分層儲存之叢集的預設值
unclean.leader.election.enable	指示您是否不在 ISR 集中的複本作為領導者與最後手段，即使這可能會導致資料遺失。	true	false
zookeeper.session.timeout.ms	Apache ZooKeeper 工作階段逾時 (以毫秒計)。	18000	18000
zookeeper.set.acl	設定使用安全 ACL的用戶端。	false	false

如需有關指定自訂組態值的詳細資訊，請參閱the section called “自訂組態”。

分層儲存主題層級組態的準則

以下是您在主題層級設定分層儲存時的預設設定和限制。

- 對於已啟用分層儲存的主題，Amazon MSK 不支援較小的日誌區段大小。如果您要建立區段，最小日誌區段大小為 48 MiB，或最小區段滾動時間為 10 分鐘。這些值會對應 segment.bytes 和 segment.ms 屬性。
- local.retention.ms/bytes 的值不能等於或超過 retention.ms/bytes 的值。這是分層儲存保留設定。
- local.retention.ms/bytes 的預設值為 -2。這意味著 retention.ms 的值會用於 local.retention.ms/bytes。在這種情況下，資料會同時保留在本機儲存和分層儲存 (每個儲存中一個副本) 中，而且會同時過期。對於此選項，本機資料的副本會保留至遠端儲存。在這種情況下，從取用流量讀取的資料來自本地儲存。
- retention.ms 的預設值為 7 天。retention.bytes 沒有預設大小限制。
- retention.ms/bytes 的最小值為 -1。這意味著無限保留。
- local.retention.ms/bytes 的最小值為 -2。這意味著在本地儲存中無限保留。它與 retention.ms/bytes 設定為 -1 時相符。
- 對於已啟動分層儲存的主題，必須設定主題層級組態 retention.ms 的值。retention.ms 最小值為 3 天。

Amazon MSK 組態操作

此主題說明如何建立自訂 MSK 組態，以及如何對其執行操作。如需如何使用 MSK 組態來建立或更新叢集的資訊，請參閱 運作方式。

本主題包含下列章節：

- 建立 MSK 組態
- 更新 MSK 組態
- 刪除 MSK 組態
- 描述 MSK 組態
- 描述 MSK 組態修訂
- 列出帳戶內目前區域的所有 MSK 組態

建立 MSK 組態

1. 建立一個檔案，在其中指定欲設定的組態屬性及欲指派的值。以下為範例組態檔案的內容。

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

log.roll.ms = 604800000
```

2. 執行下列 AWS CLI 命令，並將 *config-file-path* 取代為您在上一個步驟中儲存組態的檔案路徑。

> **Note**
>
> 您為組態選擇的名稱必須符合以下規則運算式："^[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
```

以下是執行此命令後成功回應的範例。

```
{
```

```
    "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. 上一個命令會傳回新組態的 Amazon Resource Name (ARN)。儲存此 ARN，因為其他命令須用其來參照此組態。若您遺失組態 ARN，可列出帳戶內的所有組態來找到它。

更新 MSK 組態

1. 建立一個檔案，在其中指定欲更新的組態屬性及欲指派的值。以下為範例組態檔案的內容。

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

min.insync.replicas = 2
```

2. 執行下列 AWS CLI 命令，使用您在上一步中儲存組態的檔案路徑取代 *config-file-path*。

   使用建立組態時取得的 ARN 取代 *configuration-arn*。若您建立組態時未儲存 ARN，則可使用 `list-configurations` 命令來列出帳戶內的所有組態。清單中列出的所需組態會顯示在回應中。組態 ARN 也會出現於該清單中。

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

3. 以下是執行此命令後成功回應的範例。

```
{
    "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
    }
```

```
}
```

刪除 MSK 組態

下列程序說明如何刪除未連接至叢集的組態。您無法刪除連接至叢集的組態。

1. 若要執行此範例，請使用建立組態時取得的 ARN 取代 *configuration-arn*。若您建立組態時未儲存 ARN，則可使用 `list-configurations` 命令來列出帳戶內的所有組態。清單中列出的所需組態會顯示在回應中。組態 ARN 也會出現於該清單中。

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

2. 以下是執行此命令後成功回應的範例。

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

描述 MSK 組態

1. 下列命令會傳回組態的中繼資料。若要取得組態的詳細描述，請執行 `describe-configuration-revision`。

   若要執行此範例，請使用建立組態時取得的 ARN 取代 *configuration-arn*。若您建立組態時未儲存 ARN，則可使用 `list-configurations` 命令來列出帳戶內的所有組態。清單中列出的所需組態會顯示在回應中。組態 ARN 也會出現於該清單中。

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

2. 以下是執行此命令後成功回應的範例。

   ```
   {
       "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 組態修訂

如果您使用 `describe-configuration` 命令來描述 MSK 組態，您會看到組態的中繼資料。若要取得組態的描述，請改用此命令：`describe-configuration-revision`。

- 執行下列命令，請使用建立組態時取得的 ARN 取代 *`configuration-arn`*。若您建立組態時未儲存 ARN，則可使用 `list-configurations` 命令來列出帳戶內的所有組態。清單中列出的所需組態會顯示在回應中。組態 ARN 也會出現於該清單中。

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

  以下是執行此命令後成功回應的範例。

  ```
  {
      "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` 的值採用 Base64 編碼。若您使用 Base64 解碼器 (如 https://www.base64decode.org/) 來手動解碼，則會取得用來建立自訂組態的原始組態檔案內容。在此情況下，您會取得下列內容：

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

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

列出帳戶內目前區域的所有 MSK 組態

- 執行下列命令。

```
aws kafka list-configurations
```

以下是執行此命令後成功回應的範例。

```
{
    "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 可在美國東部 (俄亥俄)、美國東部 (維吉尼亞北部)、美國西部 (奧勒岡)、加拿大 (中部)、亞太區域 (孟買)、亞太區域 (新加坡)、亞太區域 (雪梨)、亞太區域 (東京)、亞太區域 (首爾)、歐洲 (法蘭克福)、歐洲 (斯德哥爾摩)、歐洲 (愛爾蘭)、歐洲 (巴黎) 和歐洲 (倫敦) 區域使用。

MSK Serverless 是 Amazon MSK 的叢集類型，可讓您執行 Apache Kafka 而無須管理和擴展叢集容量。它會自動佈建和擴展容量，同時管理您主題中的分區，因此您可以串流資料，而無須考量適當調整叢集大小或調整叢集的規模。MSK Serverless 提供以輸出量為基礎的定價模式，因此您只需按實際用量付費。若您的應用程式需要可自動擴展和縮減的隨需串流容量，請考量使用無伺服器叢集。

MSK Serverless 與 Apache Kafka 完全相容，因此您可以使用任何相容的用戶端應用程式來產生和取用資料。它還與下列服務整合：

- AWS PrivateLink 提供私人連接
- AWS Identity and Access Management （ IAM ）用於使用 Java 和非 Java 語言的身份驗證和授權。如需設定 IAM 用戶端的指示，請參閱 設定 IAM 存取控制的用戶端。
- AWS Glue 綱要管理的綱要登錄
- Amazon Managed Service for Apache Flink (用於以 Apache Flink 為基礎的串流處理)
- AWS Lambda 用於事件處理

> **Note**
>
> MSK Serverless 需要所有叢集的 IAM 存取控制。不支援 Apache Kafka 存取控制清單 (ACL)。如需詳細資訊，請參閱 the section called “IAM 存取控制”。
> 如需有關適用於 MSK Serverless 的服務配額相關資訊，請參閱 the section called “無伺服器叢集的配額”。

若要開始使用無伺服器叢集，並進一步了解無伺服器叢集的組態和監控選項，請參閱下列內容。

主題

開始使用 MSK Serverless 叢集

本教學課程將說明如何建立 MSK Serverless 叢集、建立可存取叢集的用戶端機器、使用用戶端在叢集上建立主題，以及將資料寫入這些主題的相關範例。此演練不代表您在建立無伺服器叢集時可選擇的所有選項。為求簡化起見，我們在此演練的不同部分中選擇預設選項。這並不表示只有這些選項才能用於設定無伺服器叢集。您也可以使用 AWS CLI 或 Amazon MSK API。如需詳細資訊，請參閱 Amazon MSK API Reference 2.0。

主題

步驟 1：建立 MSK Serverless 叢集

在此步驟中，您會執行兩項任務。首先使用預設設定建立 MSK Serverless 叢集。接著則是收集叢集的相關資訊。這是您在稍後步驟中建立可傳送資料至叢集的用戶端時所需的資訊。

建立無伺服器叢集

1. 登入 AWS Management Console，然後開啟 Amazon MSK 主控台，網址為 https://console.aws.amazon.com/msk/home。
2. 選擇建立叢集。
3. 針對建立方法，請確認已選取快速建立選項。快速建立選項可讓您使用預設設定建立無伺服器叢集。
4. 在叢集名稱中，輸入描述性名稱，例如 **msk-serverless-tutorial-cluster**。
5. 針對一般叢集屬性，請選擇無伺服器作為叢集類型。針對剩餘的一般叢集屬性，使用預設值。

6. 留意所有叢集設定下方的表格。此表格會列出重要設定 (例如網路和可用性) 的預設值，並指出您是否可在建立叢集後變更每個設定值。若要在建立叢集之前變更設定，請選擇建立方法下的自訂建立選項。

> **Note**
>
> 您可以透過 MSK Serverless 叢集，從最多五個不同的 VPC 連線用戶端。若要協助用戶端應用程式在中斷時切換至其他可用區域，您必須在每個 VPC 中至少指定兩個子網路。

7. 選擇建立叢集。

收集叢集的相關資訊

1. 在叢集摘要區段中，選擇檢視用戶端資訊。在 Amazon MSK 完成叢集建立作業之前，此按鈕會保持灰色。您可能需要靜待數分鐘，直到按鈕可供點選為止。
2. 複製端點標籤下的字串。此為您的引導伺服器字串。
3. 選擇屬性索引標籤。
4. 在網路設定區段下，複製並儲存子網路和安全群組的 ID，稍後建立用戶端機器時會需要此資訊。
5. 選擇任一子網路。這會開啟 Amazon VPC 主控台。尋找與子網路關聯之 Amazon VPC 的 ID。儲存此 Amazon VPC ID 以供後續使用。

後續步驟

步驟 2：建立 IAM 角色

步驟 2：建立 IAM 角色

在此步驟中，您會執行兩項任務。第一項任務是建立 IAM 政策，用於授予在叢集上建立主題，並將資料傳送至這些主題的存取權限。第二項任務是建立 IAM 角色，並將此政策與該角色建立關聯。在稍後的步驟中，我們會建立擔任此角色的用戶端機器，使用它在叢集上建立主題，並將資料傳送至該主題。

建立能夠建立和寫入主題的 IAM 政策

1. 開啟位於 https://console.aws.amazon.com/iam/ 的 IAM 主控台。
2. 在導覽窗格中，選擇政策。
3. 選擇建立政策。
4. 選擇 JSON 索引標籤，然後使用下列 JSON 取代編輯器視窗中的 JSON。

使用您建立叢集所在的 AWS 區域 代碼取代 *region*。使用您的帳戶 ID 取代 *Account-ID*。*msk-serverless-tutorial-cluster*以無伺服器叢集的名稱取代。

```
{
    "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/*"
            ]
        }
    ]
```

```
}
```

如需有關安全政策撰寫方式的相關說明，請參閱 the section called “IAM 存取控制”。

5. 選擇下一步：標籤。
6. 選擇下一步：檢閱。
7. 針對政策名稱，請輸入描述性名稱，例如 **msk-serverless-tutorial-policy**。
8. 選擇建立政策。

建立 IAM 角色，並將政策連接至該角色

1. 在導覽窗格中，選擇角色。
2. 選擇建立角色。
3. 在一般使用案例下，選擇 EC2，然後選擇下一步：許可。
4. 在搜尋方塊中，輸入您先前為此教學課程建立的政策名稱。然後選取政策左側的核取方塊。
5. 選擇下一步：標籤。
6. 選擇下一步：檢閱。
7. 針對角色名稱，請輸入描述性名稱，例如 **msk-serverless-tutorial-role**。
8. 選擇建立角色。

後續步驟

步驟 3：建立用戶端機器

步驟 3：建立用戶端機器

在此步驟中，您會執行兩項任務。第一項任務是建立一個 Amazon EC2 執行個體作為 Apache Kafka 用戶端機器。第二項任務是在機器上安裝 Java 和 Apache Kafka 工具。

建立用戶端機器

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 選擇啟動執行個體。
3. 輸入用戶端機器的描述性名稱，例如 **msk-serverless-tutorial-client**。
4. 確認已選取 Amazon Linux 2 AMI (HVM) – Kernel 5.10，SSD 磁碟區類型 作為 Amazon Machine Image (AMI) 類型。

5. 確認選取 t2.micro 執行個體類型。
6. 在金鑰對 (登入) 下，選擇建立新金鑰對。輸入 **MSKServerlessKeyPair** 作為金鑰對名稱。選擇下載金鑰對。或者，您也可以使用現有的金鑰對。
7. 針對網路設定，選擇編輯。
8. 在 VPC 下，輸入無伺服器叢集的虛擬私有雲端 (VPC) ID。此為以 Amazon VPC 服務為基礎的 VPC，您在建立叢集後會儲存其 ID。
9. 針對子網路，選擇您在建立叢集後儲存其 ID 的子網路。
10. 針對防火牆 (安全群組)，選取與叢集關聯的安全群組。若該安全群組具有允許流量從安全群組傳輸至自身的傳入規則，此值會運作。使用此規則後，同個安全群組的成員可彼此通訊。如需詳細資訊，請參閱《Amazon VPC 開發人員指南》中的安全群組規則。
11. 展開進階詳細資料區段，然後選擇您在 步驟 2：建立 IAM 角色 中建立的 IAM 角色。
12. 選擇啟動。
13. 在左側導覽窗格中，選擇執行個體。然後在代表新建立 Amazon EC2 執行個體的列中，選擇核取方塊。自此之後，我們會將此執行個體稱為用戶端機器。
14. 選擇連線，然後遵循指示連線至用戶端機器。

在用戶端機器上設定 Apache Kafka 用戶端工具

1. 如要安裝 Java，請在用戶端機器上執行下列命令：

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

2. 如要取得建立主題與傳送資料所需的 Apache Kafka 工具，請執行下列命令：

```
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. 前往 `kafka_2.12-2.8.1/libs` 目錄，然後執行下列命令以下載 Amazon MSK IAM JAR 檔案。Amazon MSK IAM JAR 可讓用戶端機器存取叢集。

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

4. 前往 `kafka_2.12-2.8.1/bin` 目錄。複製下列屬性設定，並將其貼入新檔案。將檔案命名為 `client.properties` 並儲存。

```
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
```

後續步驟

步驟 4：建立 Apache Kafka 主題

步驟 4：建立 Apache Kafka 主題

在此步驟中，您可以使用先前建立的用戶端機器，在無伺服器叢集上建立主題。

建立主題並在其中寫入資料

1. 在下列 export 命令中，使用建立叢集之後儲存的引導伺服器字串取代 *my-endpoint*。接著前往用戶端機器上的 kafka_2.12-2.8.1/bin 目錄，執行 export 命令。

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

2. 執行下列命令，建立名為 msk-serverless-tutorial 的主題。

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

後續步驟

步驟 5：產生和取用資料

步驟 5：產生和取用資料

在此步驟中，您會使用在上一個步驟建立的主題來產生和取用資料。

產生和取用訊息

1. 執行下列命令以建立主控台生產者。

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

2. 輸入您想要的任何訊息，然後按 Enter 鍵。重複此步驟兩次或三次。每次輸入一行，然後按 Enter 鍵，該行會作為獨立訊息傳送至您的叢集。
3. 保持與用戶端機器的連線開啟，然後開啟第二個並在新視窗中與該機器單獨連線。
4. 執行下列命令，使用第二個用戶端機器連線建立主控台取用者。使用建立叢集之後儲存的引導伺服器字串取代 *my-endpoint*。

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

   當您使用主控台生產者命令時，您會開始看到先前輸入的訊息。
5. 在生產者視窗中輸入更多訊息，並觀看它們出現在取用者視窗中。

後續步驟

步驟 6：刪除資源

步驟 6：刪除資源

在此步驟中，刪除您在本教學課程中建立的資源。

刪除叢集

1. 開啟位於 https://console.aws.amazon.com/msk/home 的 Amazon MSK 主控台。
2. 在叢集清單中，選擇您要為此教學課程建立的叢集。
3. 針對動作，選擇刪除叢集。
4. 在欄位中輸入 `delete`，然後選擇刪除。

停止用戶端機器

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在 Amazon EC2 執行個體清單中，選擇您為本教學課程建立的用戶端機器。
3. 依序選擇執行個體狀態和終止執行個體。

4. 選擇終止。

刪除 IAM 政策和角色

1. 前往 https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。
2. 在導覽窗格中，選擇角色。
3. 在搜尋方塊中，輸入您為本教學課程建立的 IAM 角色名稱。
4. 選擇角色。然後選擇刪除角色並確認刪除。
5. 在導覽窗格中，選擇政策。
6. 在搜尋方塊中，輸入您為本教學課程建立的政策名稱。
7. 選擇政策以開啟其摘要頁面。在政策的摘要頁面上，選擇刪除政策。
8. 選擇刪除。

無伺服器叢集的組態

Amazon MSK 會為無伺服器叢集設定代理程式組態屬性。您無法變更這些代理程式組態屬性設定。不過，您可以設定下列主題組態屬性。

組態屬性	預設	是否可編輯	允許值上限
cleanup.policy	Delete	可以，但僅限於主題建立時	
compression.type	生產者	是	
max.message.bytes	1048588	是	8 MiB
message.timestamp.difference.max.ms	long.max	是	
message.timestamp.type	CreateTime	是	
retention.bytes	250 GiB	是	250 GiB
retention.ms	7 天	是	無限制

您也可以使用 Apache Kafka 命令，設定或修改新的或現有主題的主題層級組態屬性。如需主題層級組態屬性及其設定範例的詳細資訊，請參閱官方 Apache Kafka 文件中的 Topic-Level Configs。

監控無伺服器叢集

Amazon MSK 與 Amazon 整合， CloudWatch 因此您可以收集、檢視和分析 MSK 無伺服器叢集的指標。下表中顯示的指標適用於所有無伺服器叢集。由於這些指標是以主題中每個分區的個別資料點形式發布，建議您以 'SUM' 統計資料檢視它們，以取得主題層級檢視。

Amazon MSK 以 CloudWatch 每分鐘一次的頻率發佈PerSec指標。這表示一分鐘期間的 'SUM' 統計資料可精確地代表 PerSec 指標的每秒資料。若要在超過一分鐘的時間內收集每秒資料，請使用下列 CloudWatch 數學運算式：m1 * 60/PERIOD(m1)。

DEFAULT 監控層級提供的指標

名稱	可見時	維度	描述
BytesInPerSec	生產者寫入主題之後	叢集名稱、主題	從用戶端接收的每秒位元組數量。此指標適用於每個主題。
BytesOutPerSec	取用者群組取用一個主題之後	叢集名稱、主題	傳送至用戶端的每秒位元組數量。此指標適用於每個主題。
FetchMessageConversionsPerSec	取用者群組取用一個主題之後	叢集名稱、主題	主題的每秒擷取訊息轉換次數。
EstimatedMaxTimeLag	取用者群組取用一個主題之後	叢集名稱、取用者群組、主題	MaxOffsetLag 測量結果的時間估計。
MaxOffsetLag	取用者群組取用一個主題之後	叢集名稱、取用者群組、主題	主題中所有分區的最大偏移延遲。
MessagesInPerSec	生產者寫入主題之後	叢集名稱、主題	主題每秒傳入訊息的數量。

名稱	可見時	維度	描述
ProduceMessageConversionsPerSec	生產者寫入主題之後	叢集名稱、主題	主題的每秒產生訊息轉換次數。
SumOffsetLag	取用者群組取用一個主題之後	叢集名稱、取用者群組、主題	主題中所有分區的彙整偏移延遲。

檢視 MSK Serverless 指標

1. 請登入 AWS Management Console 並開啟 CloudWatch 主控台，網址為 https://console.aws.amazon.com/cloudwatch/。
2. 在導覽窗格的指標下方，選擇所有指標。
3. 在指標中搜尋 **kafka** 詞語。
4. 選擇 AWS/Kafka/叢集名稱、主題或 AWS/Kafka/叢集名稱、取用者群組、主題，以查看不同的指標。

MSK Connect

什麼是 MSK Connect？

MSK Connect 是 Amazon MSK 的一項功能，可讓開發人員輕鬆將資料串流至或串流出 Apache Kafka 叢集。MSK Connect 會使用 Kafka Connect 2.7.1，其為開放源代碼框架，用於將 Apache Kafka 叢集與外部系統 (例如資料庫、搜尋索引和檔案系統) 相連。透過 MSK Connect，您可以部署專為 Kafka Connect 建立的全受管連接器，將資料移入 Amazon S3 和 Amazon 服務等熱門資料存放區或從熱門資料存放區提取資料。 OpenSearch 您可以部署由第三方 (例如 Debezium) 開發的連接器，用於將變更日誌從資料庫串流至 Apache Kafka 叢集，或在不需變更程式碼的情況下部署現有連接器。連接器會自動擴展以適應負載的變化，您僅需按照實際使用的資源量付費。

使用來源連接器將資料從外部系統匯入至您的主題中。您可以使用目的地連接器將主題中的資料匯出至外部系統。

MSK Connect 會利用與 Amazon VPC 的連線來支援 Apache Kafka 叢集的連接器，無論叢集是 MSK 叢集還是獨立託管的 Apache Kafka 叢集皆可。

MSK Connect 會持續監控連接器運作狀態和傳送狀態、修補程式和管理基礎硬體，並自動擴展連接器以符合輸送量的變更。

若要開始使用 MSK Connect，請參閱 the section called “開始使用”。

若要瞭解您可以使用 MSK Connect 建立的 AWS 資源，請參閱the section called “連接器”the section called “外掛程式”、和the section called “工作程序”。

如需有關 MSK Connect API 的相關資訊，請參閱 Amazon MSK Connect API Reference。

開始使用 MSK Connect

step-by-step 本教學課程使用建立 MSK 叢集和接收器連接器，將資料從叢集傳送至 S3 儲存貯體。AWS Management Console

主題

- 步驟 1：設定必要資源
- 步驟 2：建立自訂外掛程式

步驟 1：設定必要資源

在此步驟中，您需要建立以下此開始使用案例中所需的資源：

- S3 儲存貯體，作為從連接器接收資料的目的地。
- MSK 叢集，作為接收傳送資料的目的地。接著，連接器會從此叢集讀取資料，並將其傳送至目的地 S3 儲存貯體。
- IAM 角色，允許連接器寫入目的地 S3 儲存貯體。
- Amazon VPC 端點，讓您可從具有叢集和連接器的 Amazon VPC 將資料傳送至 Amazon S3。

建立 S3 儲存貯體

1. 登入 AWS Management Console 並開啟 Amazon S3 主控台，網址為 https://console.aws.amazon.com/s3/。
2. 選擇建立儲存貯體。
3. 請為儲存貯體名稱輸入描述性名稱，例如 `mkc-tutorial-destination-bucket`。
4. 向下捲動並選擇建立儲存貯體。
5. 在儲存貯體清單中，選擇您新建立的儲存貯體。
6. 選擇 Create folder (建立資料夾)。
7. 輸入 `tutorial` 作為資料夾名稱，然後向下捲動並選擇建立資料夾。

建立叢集

1. 開啟 Amazon MSK 主控台，網址為 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。
2. 在左窗格的 MSK 叢集下，選擇叢集。
3. 選擇建立叢集。
4. 選擇自訂建立。
5. 針對叢集名稱，請輸入 `mkc-tutorial-cluster`。

6. 在「一般叢集屬性」下，請選擇佈建作為叢集類型。
7. 在聯網下，請選擇 Amazon VPC。然後請選取您要使用的可用區域和子網路。請記住您選取的 Amazon VPC 和子網路的 ID，因為稍後在本教學課程中您將需要這些 ID。
8. 在存取控制方法下，請確認僅選取未身分驗證的存取。
9. 在加密下，請確認僅選取純文字。
10. 繼續執行精靈，然後選擇建立叢集。這會帶您前往叢集的詳細資訊頁面。在該頁面的已套用的安全群組下，找到安全群組 ID。記住該 ID，因為稍後在本教學可課程中您將需要該 ID。

建立可寫入目的地儲存貯體的IAM 角色

1. 開啟位於 https://console.aws.amazon.com/iam/ 的 IAM 主控台。
2. 在左窗格的存取管理下，選擇角色。
3. 選擇建立角色。
4. 在或者選取服務以檢視其使用案例下，選擇 S3。
5. 向下捲動並在選取您的使用案例下方，再次選擇 S3。
6. 選擇 Next: Permissions (下一步：許可)。
7. 選擇建立政策。這會在您的瀏覽器中開啟新索引標籤，您將在其中建立政策。請持續開啟原始角色建立索引標籤，因為我們稍後會回到該索引標籤。
8. 選擇 JSON 索引標籤，並使用以下政策取代視窗中的文字。

```
{
  "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. 選擇下一步：標籤。
10. 選擇下一步：檢閱。
11. 輸入 mkc-tutorial-policy 作為政策名稱，然後向下滑動並選擇建立政策。
12. 返回您正在其中建立角色的瀏覽器索引標籤，然後選擇重新整理按鈕。
13. 找到 mkc-tutorial-policy 並選擇其左側的按鈕以進行選取。
14. 選擇下一步：標籤。
15. 選擇下一步：檢閱。
16. 輸入 mkc-tutorial-role 作為角色名稱，然後刪除描述方塊中的文字。
17. 選擇建立角色。

允許 MSK Connect 擔任該角色

1. 在 IAM 主控台中，於左窗格的存取管理下，選擇角色。
2. 找到 mkc-tutorial-role 並選擇。
3. 在該角色的摘要下，選擇信任關係索引標籤。
4. 選擇編輯信任關係。
5. 使用以下 JSON 來取代現有信任政策。

```
{
```

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

6. 選擇 Update Trust Policy (更新信任政策)。

建立從叢集的 VPC 到 Amazon S3 的 Amazon VPC 端點

1. 前往 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在左側窗格中選擇端點。
3. 選擇建立端點。
4. 在服務名稱下，選擇 com.amazonaws.us-east-1.s3 服務和閘道類型。
5. 選擇叢集的 VPC，然後選取與叢集子網路相關聯之路由表左側的方塊。
6. 選擇建立端點。

後續步驟

步驟 2：建立自訂外掛程式

步驟 2：建立自訂外掛程式

外掛程式包含定義連接器邏輯的程式碼。在此步驟中，您會建立具有 Lenses Amazon S3 目的地連接器程式碼的自訂外掛程式。在稍後的步驟中，當您建立 MSK 連接器時，您需說明其程式碼位於此自訂外掛程式中。您可以使用相同外掛程式來建立具有不同組態的多個 MSK 連接器。

建立自訂外掛程式

1. 下載 S3 連接器。
2. 上傳 ZIP 檔案至您有權存取的 S3 儲存貯體。如需有關如何將檔案上傳到 Amazon S3 的詳細資訊，請參閱《Amazon S3 使用者指南》中的上傳物件。

3. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
4. 在左窗格中展開 MSK Connect，然後選擇自訂外掛程式。
5. 選擇建立自訂外掛程式。
6. 選擇 Browse S3 (瀏覽 S3)。
7. 在儲存貯體清單中，找到您上傳 ZIP 檔案的儲存貯體，然後選擇該儲存貯體。
8. 在儲存貯體內的物件清單中，選取 ZIP 檔案左側的選項按鈕，然後選擇標示為選擇的按鈕。
9. 輸入 `mkc-tutorial-plugin` 作為自訂外掛程式名稱，然後選擇建立自訂外掛程式。

完成創建自定義插件可能需 AWS 要幾分鐘的時間。在建立程序完成後，您會在瀏覽器視窗頂端的橫幅中看到以下訊息。

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

後續步驟

步驟 3：建立用戶端機器和 Apache Kafka 主題

步驟 3：建立用戶端機器和 Apache Kafka 主題

在此步驟中，您會建立 Amazon EC2 執行個體，以用作 Apache Kafka 用戶端執行個體。然後，您可使用此執行個體在叢集上建立主題。

建立用戶端機器

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 選擇啟動執行個體。
3. 輸入用戶端機器的名稱，例如 **`mkc-tutorial-client`**。
4. 確認已選擇 Amazon Linux 2 AMI (HVM) – Kernel 5.10，SSD 磁碟區類型作為 Amazon Machine Image (AMI) 類型。
5. 選擇 t2.xlarge 執行個體類型。
6. 在金鑰對 (登入) 下，選擇建立新金鑰對。輸入 **`mkc-tutorial-key-pair`** 作為金鑰對名稱，然後選擇下載金鑰對。或者，您也可以使用現有的金鑰對。
7. 選擇啟動執行個體。

8. 選擇檢視執行個體。然後，在安全群組資料欄中，選擇與新執行個體相關聯的安全群組。複製並儲存安全群組的 ID，以供日後使用。

允許新建立的用戶端將資料傳送至叢集

1. 前往 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在左窗格的安全下，選擇安全群組。在安全群組 ID 資料欄中，尋找叢集的安全群組。在 the section called “步驟 1：設定必要資源” 中建立叢集後，便已儲存此安全群組的 ID。請選取安全群組該列左側的方塊，以選擇此安全群組。請確認並未同時選取其他安全群組。
3. 在畫面下半部中，選擇傳入規則索引標籤。
4. 選擇 Edit inbound Rules (編輯傳入規則)。
5. 在畫面左下方，選擇新增規則。
6. 在新規則中，於類型資料欄中選擇所有流量。在來源資料欄的右側欄位中，輸入用戶端機器安全群組的 ID。這是您在建立用戶端機器後儲存的安全群組 ID。
7. 選擇儲存規則。MSK 叢集現在可接受您在上一個程序中建立之用戶端的所有流量。

若要建立主題

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在執行個體表格中選擇 `mkc-tutorial-client`。
3. 在畫面頂端附近選擇連線，然後依照指示連線至執行個體。
4. 執行以下命令，在用戶端執行個體上安裝 Java：

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

5. 執行下列命令下載 Apache Kafka。

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

> **Note**
>
> 如果您想要使用此命令中以外的鏡像網站，您可以在 Apache 網站上選擇不同的鏡像網站。

6. 在您在先前步驟中下載 TAR 檔案的目錄中執行下列命令。

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

7. 前往 kafka_2.12-2.2.1 目錄。
8. 開啟 Amazon MSK 主控台，網址為 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。
9. 在左窗格中選擇叢集，然後選擇名稱 mkc-tutorial-cluster。
10. 選擇檢視用戶端資訊。
11. 複製純文字連線字串。
12. 選擇完成。
13. 在用戶端執行個體 (mkc-tutorial-client) 上執行下列命令，並以檢視叢集用戶端資訊時儲存的值取代*bootstrapServerString*。

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

如果命令成功，您會看到以下訊息：Created topic mkc-tutorial-topic.

後續步驟

步驟 4：建立連接器

步驟 4：建立連接器

建立連接器

1. 登入 AWS Management Console，然後開啟 Amazon MSK 主控台，網址為 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。
2. 在左窗格中展開 MSK Connect，然後選擇連接器。
3. 選擇 Create connector (建立連接器)。
4. 在外掛程式清單中，選擇 mkc-tutorial-plugin，然後選擇下一步。
5. 針對連接器名稱，請輸入 mkc-tutorial-connector。
6. 在叢集清單中，選擇 mkc-tutorial-cluster。
7. 複製以下組態並貼入連接器組態欄位。

```
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. 在存取許可下選擇 mkc-tutorial-role。
9. 選擇下一步。在安全頁面上，再次選擇下一步。
10. 在日誌頁面上，選擇下一步。
11. 在檢閱和建立下選擇建立連接器。

後續步驟

步驟 5：傳送資料

步驟 5：傳送資料

在此步驟中，您會將資料傳送至先前建立的 Apache Kafka 主題，然後在目的地 S3 儲存貯體中尋找同一項資料。

傳送資料至 MSK 叢集

1. 在用戶端執行個體上安裝 Apache Kafka 的 bin 資料夾中，建立一個名為 client.properties 且具有以下內容的文字檔案。

```
security.protocol=PLAINTEXT
```

2. 執行下列命令以建立主控台生產者。取代*BootstrapBrokerString*為執行上一個指令時取得的值。

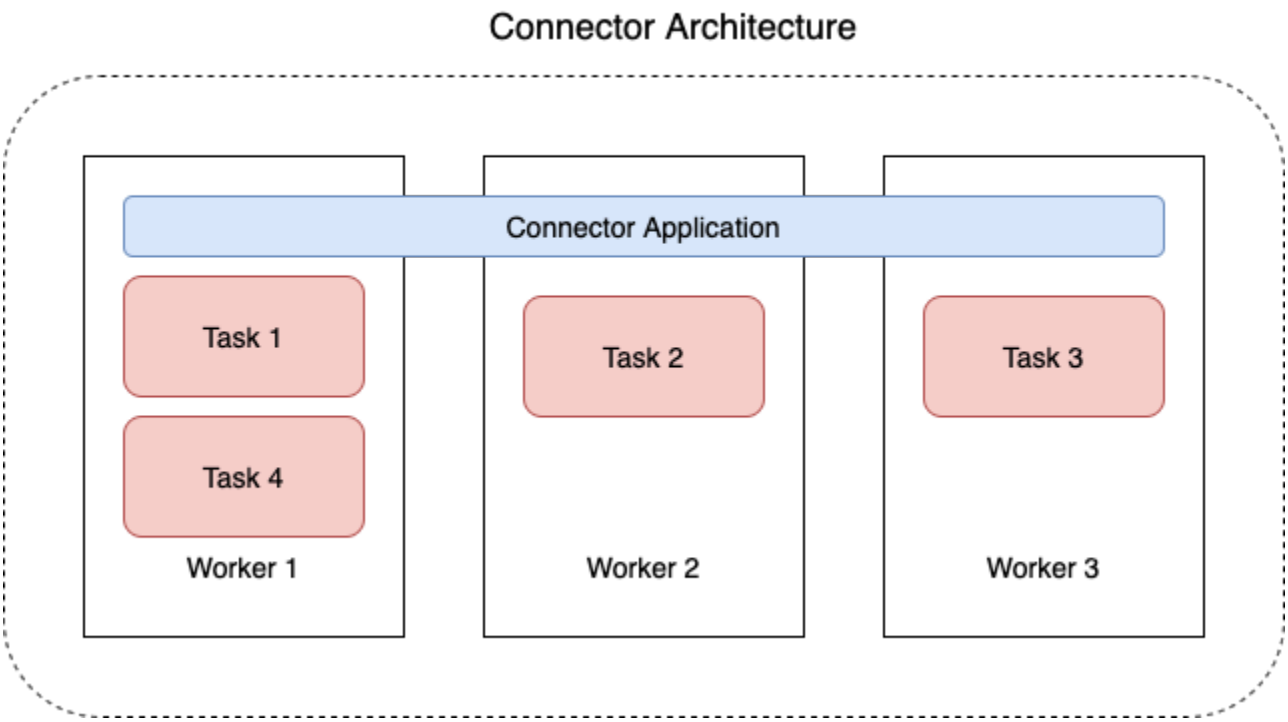
```
<path-to-your-kafka-installation>/bin/kafka-console-producer.sh --broker-
list BootstrapBrokerString --producer.config client.properties --topic mkc-
tutorial-topic
```

3. 輸入您想要的任何訊息，然後按 Enter 鍵。重複此步驟兩次或三次。每次輸入一行，然後按 Enter，該行會作為單獨訊息傳送到您的 Apache Kafka 叢集中。
4. 在目的地 Amazon S3 儲存貯體中查看，尋找您在上一個步驟中傳送的訊息。

連接器

連接器會將外部系統和 Amazon 服務與 Apache Kafka 整合，方法是持續將資料來源的串流資料複製到 Apache Kafka 叢集中，或持續將叢集中的資料複製到資料目的地中。在將資料傳送至目的地之前，連接器也可執行輕量型邏輯，例如轉換、格式轉換或篩選資料。來源連接器會從資料來源提取資料，並將此資料推送至叢集中，同時，目的地連接器則會從叢集提取資料，並將此資料推送至資料目的地中。

以下圖表說明連接器的架構。工作程序是執行連接器邏輯的 Java 虛擬機器 (JVM) 程序。每個工作程序皆會建立一組在平行執行緒中執行的任務，並執行複製資料的工作。任務不會存放狀態，因此可以隨時啟動、停止或重新啟動，以提供彈性且可擴展的資料管道。



連接器容量

連接器的總容量將依該連接器具有的工作程序數量，以及每個工作程序的 MSK Connect 單位 (MCU) 數量而定。一個 MCU 代表 1 個 vCPU 的運算和 4 GiB 的記憶體。MCU 記憶體與工作程序執行個體的總記憶體有關，而非與使用中的堆積記憶體相關。

MSK Connect 工作者會使用客戶提供的子網路中的 IP 位址。每個 Worker 都會使用其中一個客戶提供的子網路中的一個 IP 位址。您應該確保提供的子網路中有足夠的 CreateConnector 可用 IP 位址來說明其指定容量，尤其是當 Worker 數量可能會波動的自動調度資源連接器時。

若要建立連接器，您必須選擇以下兩種容量模式之一。

- 佈建 - 若您知道連接器的容量需求，請選擇此模式。您可以指定兩個值：
  - 工作程序數量。
  - 每個工作程序的 MCU 數目。
- 自動擴展 - 若連接器的容量需求可變，或者您事先不知道容量需求，請選擇此模式。在您使用自動擴展模式時，Amazon MSK Connect 會使用與連接器中執行的工作程序數量以及每個工作程序的 MCU 數量成比例的值，來覆寫連接器的 `tasks.max` 屬性。

  您可以指定三組值：
  - 工作程序數量下限和上限。
  - CP 使用率的縮減和橫向擴展百分比，依據 `CpuUtilization` 指標決定。在連接器的 `CpuUtilization` 指標超過橫向擴展百分比時，MSK Connect 會增加連接器中執行的工作程序數量。在 `CpuUtilization` 指標低於縮減百分比時，MSK Connect 會減少工作程序的數量。工作程序的數目會永遠維持在您建立連接器時指定的數量下限和上限之間。
  - 每個工作程序的 MCU 數目。

如需有關工作程序的詳細資訊，請參閱 the section called “工作程序”。若要了解 MSK Connect 指標，請參閱 the section called “監控”。

建立連接器

使用建立連接器 AWS Management Console

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
2. 在左窗格的 MSK Connect 下，選擇連接器。
3. 選擇 Create connector (建立連接器)。
4. 您可選擇使用現有的自訂外掛程式來建立連接器，或先建立新的自訂外掛程式。如需有關自訂外掛程式以及如何建立的相關資訊，請參閱 the section called “外掛程式”。在此過程中，請假設您具有要使用的自訂外掛程式。在自訂外掛程式清單中，找到您要使用的外掛程式，然後選取其左側的方塊，然後選擇下一步。
5. 輸入名稱，以及描述 (非必要)。

6. 選擇您要連線到的叢集。
7. 指定連接器組態。您需要指定的組態參數會依據您要建立的連接器類型而定。然而，有部分參數是所有連接器都要指定的，例如 `connector.class` 和 `tasks.max` 參數。以下是 Confluent Amazon S3 目的地連接器的範例組態。

```
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. 接下來，您需設定連接器容量。您可在兩種容量模式中選擇其一：佈建和自動擴展。如需關於這兩個選項的詳細資訊，請參閱the section called “容量”。
9. 在預設工作程序組態或自訂工作程序組態中選擇其一。如需有關建立自訂工作程序組態的詳細資訊，請參閱 the section called “工作程序”。
10. 接下來，您需指定服務執行角色。這必須是 MSK Connect 可以承擔的 IAM 角色，並授予連接器存取必要 AWS 資源所需的所有權限。這些許可會依連接器的邏輯而定。如需如何建立此角色的資訊，請參閱 the section called “服務執行角色”。
11. 選擇下一步，檢閱安全性資訊，然後再次選擇下一步。
12. 指定所需的記錄選項，然後選擇下一步。如需日誌記錄的相關資訊，請參閱the section called “日誌”。
13. 選擇 Create connector (建立連接器)。

若要使用 MSK 連線 API 建立 Connect 器，請參閱CreateConnector。

外掛程式

外掛程式是包含定義連接器邏輯的程式碼的 AWS 資源。您可以將 JAR 檔案 (或包含一個或多個 JAR 檔案的 ZIP 檔案) 上傳至 S3 儲存貯體，並在建立外掛程式時指定儲存貯體的位置。在建立連接器時，

您可以指定要讓 MSK Connect 用於連接器的外掛程式。插件與連接器的關係是 one-to-many：您可以從同一個插件創建一個或多個連接器。

如需如何開發連接器程式碼的相關資訊，請參閱 Apache Kafka 文件中的 Connector Development Guide。

使用建立自訂外掛程式 AWS Management Console

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
2. 在左窗格的 MSK Connect 下選擇自訂外掛程式。
3. 選擇建立自訂外掛程式。
4. 選擇 Browse S3 (瀏覽 S3)。
5. 在 S3 儲存貯體清單中，為外掛程式選擇具有 JAR 或 ZIP 檔案的儲存貯體。
6. 在物件清單中，為外掛程式選取 JAR 或 ZIP 檔案左邊的方塊，然後選擇選擇。
7. 選擇建立自訂外掛程式。

若要使用 MSK Connect API 建立自訂外掛程式，請參閱CreateCustomPlugin。

工作程序

工作程序是執行連接器邏輯的 Java 虛擬機器 (JVM) 程序。每個工作程序皆會建立一組在平行執行緒中執行的任務，並執行複製資料的工作。任務不會存放狀態，因此可以隨時啟動、停止或重新啟動，以提供彈性且可擴展的資料管道。無論是源自於擴展事件還是非預期的失敗，其餘工作程序會自動偵測到工作程序數量的變更。它們會協調以重新平衡整組其餘工作程序的任務。Connect 工作程序會使用 Apache Kafka 的取用者群組來進行協調和重新平衡。

若連接器的容量需求可變或難以估計，您可以讓 MSK Connect 根據需要在您指定的下限和上限之間擴展工作程序數量。您也可以指定要執行連接器邏輯的工作程序確切數量。如需詳細資訊，請參閱 the section called “容量”。

MSK Connect 工作者會使用 IP 位址

MSK Connect 工作者會使用客戶提供的子網路中的 IP 位址。每個 Worker 都會使用其中一個客戶提供的子網路中的一個 IP 位址。您應該確保提供的子網路中有足夠的 CreateConnector 可用 IP 位址來說明其指定容量，尤其是當 Worker 數量可能會波動的自動調度資源連接器時。

主題

預設工作程序組態

MSK Connect 會提供以下預設工作程序組態：

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

支援的工作程序組態屬性

MSK Connect 會提供預設工作程序組態。您也可以選擇建立自訂工作程序組態，以與連接器搭配使用。以下清單列出了 Amazon MSK Connect 支援或不支援的工作程序組態屬性訊。

- `key.converter` 和 `value.converter` 屬性為必要項目。
- MSK Connect 支援以下 `producer.` 組態屬性。

```
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 支援以下 `consumer.` 組態屬性。

```
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
```

- 支援所有不以 `producer.` 或 `consumer.` 字首開頭的組態屬性，以下屬性外除外。

```
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
```

如需有關工作程序組態屬性和其代表內容的詳細資訊，請參閱 Apache Kafka 文件中的 Kafka Connect Configs。

建立自訂工作程序組態

使用建立自訂 Worker 組態 AWS Management Console

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
2. 在左窗格的 MSK Connect下，選擇工作程序組態。
3. 選擇建立工作程序組態。
4. 輸入名稱和選用描述，然後加入您要為其設定的屬性和值。
5. 選擇建立工作程序組態。

若要使用 MSK Connect API 建立背景工作設定，請參閱CreateWorkerConfiguration。

使用 `offset.storage.topic` 來管理來源連接器偏移

本節提供的資訊有助您使用偏移儲存主題來管理來源連接器偏移。偏移儲存主題是一項內部主題，Kafka Connect 會用來存放連接器和任務組態偏移。

使用預設偏移儲存主題

根據預設，Amazon MSK Connect 會為您建立的每個連接器，在 Kafka 叢集上產生新的偏移儲存主題。MSK 會使用連接器 ARN 的部分來建構預設主題名稱。例如 `__amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-4abc-8be8-c657f7e4ff32-2`。

指定自己的偏移儲存主題

若要在來源連接器之間提供偏移連續性，您可以使用所選偏移儲存主題，而非預設主題。指定偏移儲存主題有助您完成任務，例如，建立來源連接器從上一個連接器的最後一個偏移恢復讀取。

若要指定偏移儲存主題，請在建立連接器之前，在工作程序組態中提供 offset.storage.topic 屬性值。若您要重複使用偏移儲存主題來使用先前建立之連接器的偏移，您必須為新連接器指定與舊連接器相同的名稱。若您建立自訂偏移儲存主題，則必須在主題組態中將 cleanup.policy 設定為 compact。

> **Note**
>
> 若您在建立目的地連接器時指定偏移儲存主題，則若該主題尚不存在，MSK Connect 將會建立該主題。然而，該主題將不會用於儲存連接器偏移。
> 反而是使用 Kafka 取用者群組通訊協定來管理目的地連接器偏移。每個目的地連接器都會建立名為 connect-{CONNECTOR_NAME} 的群組。只要取用者群組存在，您使用相同 CONNECTOR_NAME 值建立的任何連續目的地連接器都會從上次遞交的偏移繼續執行。

Example ：指定偏移儲存主題，以使用更新後的組態重新建立來源連接器

假設您已變更資料擷取 (CDC) 連接器，且欲在不會遺失您在 CDC 串流中位置的情況下修改連接器組態。您無法更新現有的連接器組態，但可以刪除連接器，並使用相同的名稱建立新的新連接器。若要告知新連接器在 CDC 串流中開始讀取的位置，您可以在工作程序組態中指定舊連接器的偏移儲存主題。以下步驟會說明如何完成這項任務。

1. 在用戶端電腦上，執行以下命令以尋找連接器的偏移儲存主題名稱。使用您叢集的引導代理程式字串來取代 *<bootstrapBrokerString>*。如需有關取得引導代理程式字串的指示，請參閱 取得 Amazon MSK 叢集的引導代理程式。

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

   以下輸出顯示所有叢集主題的清單，包括任何預設內部連接器主題。在此範例中，現有 CDC 連接器會使用 MSK Connect 建立的預設偏移儲存主題。此即為偏移儲存主題被稱為 __amazon_msk_connect_offsets_my-mskc-connector_12345678-09e7-4abc-8be8-c657f7e4ff32-2 之原因。

   ```
   __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. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
3. 從連接器清單中選擇您的連接器。複製並儲存連接器組態欄位的內容，以便修改並使用該內容來建立新的連接器。
4. 選擇刪除以刪除連接器。文字輸入在欄位中輸入連接器名稱以確認刪除。
5. 使用符合您案例的值來建立自訂工作程序組態。如需說明，請參閱建立自訂工作程序組態。

   在工作程序組態中，您必須將之前擷取之偏移儲存主題的名稱指定為 `offset.storage.topic` 的值，如以下組態所示。

```
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**

   您必須為新連接器取名為與舊連接器相同的名稱。

   使用您在之前步驟中設定的工作程序組態來建立新的連接器。如需說明，請參閱建立連接器。

考量事項

在管理來源連接器偏移時，請考量以下事項。

- 若要指定偏移儲存主題，請提供 Kafka 主題的名稱，其連接器偏移會被儲存為工作程序組態中的 `offset.storage.topic` 值。
- 在變更連接器組態時請小心。若來源連接器會將組態中的值用於關鍵偏移記錄，則變更組態值可能會導致意外的連接器行為。我們建議您參考外掛程式的文件以取得指引。
- 自訂預設分區數量 - 除了藉由新增 `offset.storage.topic` 的方式來自訂工作程序組態之外，您還可以自訂偏移量狀態儲存主題的分區數量。內部主題的預設分區數量如下。

  - `config.storage.topic`：1，不可設定，必須為單一分區主題
  - `offset.storage.topic`：25，藉由提供 `offset.storage.partitions` 進行設定
  - `status.storage.topic`：5，藉由提供 `status.storage.partitions` 進行設定
- 手動刪除主題 - Amazon MSK Connect 會在連接器的每個部署上建立新的 Kafka 連接內部主題 (主題名稱開頭為 `__amazon_msk_connect`)。連接至已刪除連接器的舊主題不會自動受到移除，這是因為內部主題 (例如 `offset.storage.topic`) 可在連接器之間重複使用。然而，您可以手動刪除由 MSK Connect 建立但未使用的內部主題。內部主題會依 `__amazon_msk_connect_<offsets|status|configs>_`*`connector_name`*`_`*`connector_id`* 格式來命名。

  規則表達式 `__amazon_msk_connect_<offsets|status|configs>_`*`connector_name`*`_`*`connector_id`* 可用於刪除內部主題。您不應刪除執行中連接器目前正在使用的內部主題。
- 將相同名稱用於 MSK Connect 建立的內部主題 - 若您要重複使用偏移儲存主題來使用先前建立之連接器的偏移，您必須為新連接器指定與舊連接器相同的名稱。可以使用工作程序組態來設定 `offset.storage.topic` 屬性，將相同名稱指派給 `offset.storage.topic`，並在不同連接器之間重複使用。此組態在管理連接器偏移中有所描述。MSK Connect 不允許不同連接器共用 `config.storage.topic` 和 `status.storage.topic`。每次您在 MSKC 中建立新連接器時，皆會建立這些主題。它們會依 `__amazon_msk_connect_<status|configs>_`*`connector_name`*`_`*`connector_id`* 格式自動命名，因此它們的名稱在您建立的不同連接器之間也會有所不同。

使用組態供應商來外部化敏感資訊

此範例說明如何使用開放原始碼組態供應商將 Amazon MSK Connect 的敏感資訊外部化。組態供應商讓您可在連接器或工作程序組態中指定變數 (而非純文字)，而在連接器中執行的工作程序會在執行期解析這些變數。此做法可避免系統以純文字方式儲存憑證和其他秘密。範例中的組態提供者支援從 AWS Secrets Manager、Amazon S3 和 Systems Manager (SSM) 擷取組態參數。在步驟 2 中，您可以了解如何設定服務的儲存和敏感資訊擷取。

主題

步驟 1：建立自訂外掛程式並上傳至 S3

若要建立自訂外掛程式，請 msk-config-provider 透過在本機電腦上執行下列指令來建立包含連接器的 zip 檔案。

使用終端機視窗和 Debezium 作為連接器來建立自訂外掛程式

使用 AWS CLI 以超級使用者身分執行具有可讓您存取 AWS S3 儲存貯體的登入資料的命令。如需有關安裝和設定 AWS CLI 的資訊，請參閱《AWS Command Line Interface 使用者指南》中的 AWS CLI 入門。如需將 AWS CLI 與 Amazon S3 搭配使用的相關資訊，請參閱使用AWS Command Line Interface 者指南中的將 Amazon S3 搭配 AWS CLI 使用。

1. 在終端機視窗中，使用以下命令在工作區中建立名為 `custom-plugin` 的資料夾。

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

2. 使用以下命令，從 Debezium 網站下載 MySQL Connector Plug-in 的最新穩定版本。

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

   使用以下命令將下載的 gzip 文件解壓縮至 `custom-plugin` 文件夾中。

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

3. 使用以下命令來下載 MSK 組態供應商 zip 檔案。

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

   使用以下命令將下載的 zip 文件解壓縮至 `custom-plugin` 文件夾中。

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

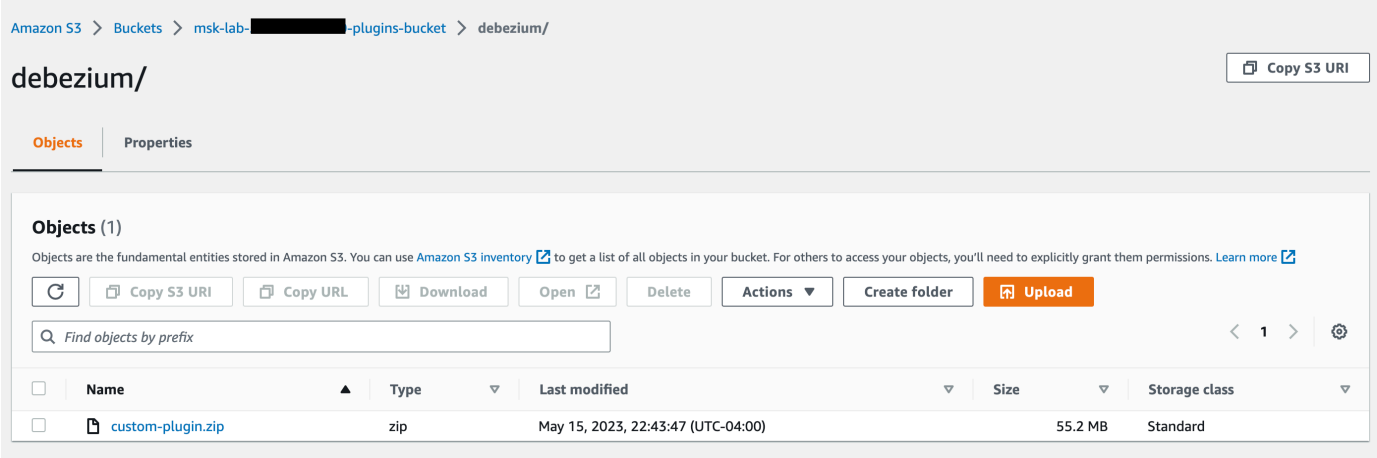
4. 將上述步驟中 MSK 組態供應商的內容和自訂連接器壓縮至名為 `custom-plugin.zip` 的單一檔案中。

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

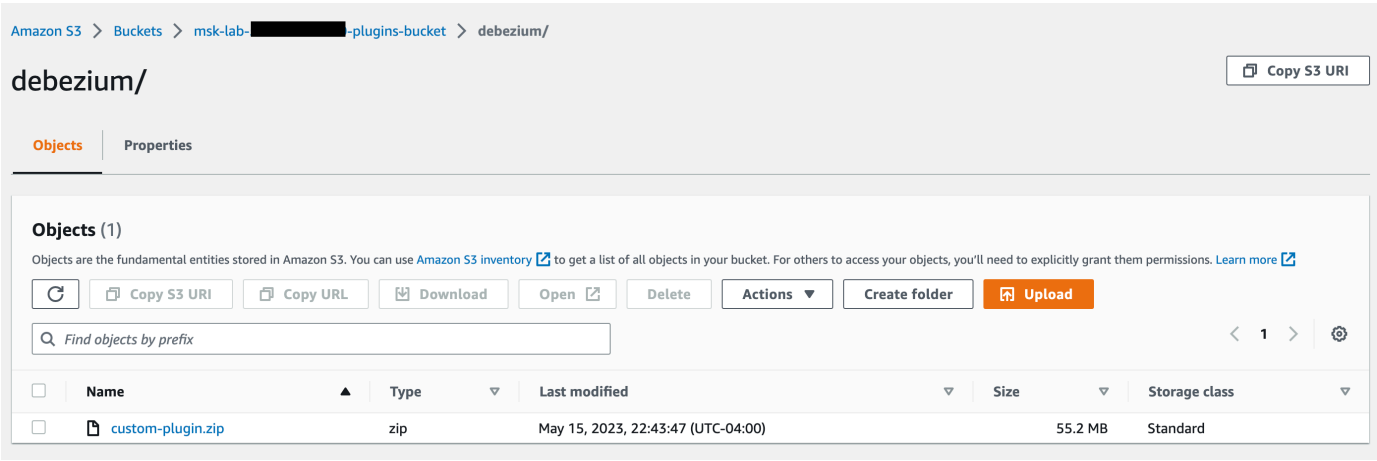
5. 將檔案上傳至 S3 以供稍後參考。

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

6. 在 Amazon MSK 主控台的 MSK Connect 區段下，選擇自訂外掛程式，然後選擇建立自訂外掛程式，並瀏覽 s3:<***S3_URI_BUCKET_LOCATION***> S3 儲存貯體，以選取您剛剛上傳的自訂外掛程式 ZIP 檔案。



7. 輸入 **debezium-custom-plugin** 作為外掛程式名稱。輸入描述 (選用)，然後選擇建立自訂外掛程式。



步驟 2：設定不同供應商的參數和許可

您可以在以下三項服務中設定參數值：

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

選取以下其中一個標籤，以取得設定該服務之參數和相關許可的指示。

Configure in Secrets Manager

在 Secrets Manager 中設定參數值

1. 開啟 Secrets Manager 主控台。
2. 建立新秘密以存放您的憑證或秘密。如需指示，請參閱《 AWS Secrets Manager 使用指南》中的「建立 AWS Secrets Manager 密碼」。
3. 複製您秘密的 ARN。
4. 將以下範例政策中的 Secrets Manager 許可新增至您的服務執行角色。替換 *<arn#AW###### ######-1#123456789000####-* 1234> 與您的秘密的 ARN。MySecret
5. 新增工作程序組態和連接器指示。

```
{
        "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. 若要使用 Secrets Manager 組態供應商，請在步驟 3 中將下列程式碼行複製到工作程序組態文字方塊：

```
# 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. 如果使用 Secrets Manager 組態供應商，請在步驟 4 中複製連接器組態的以下程式碼行。

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

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

您也可以針對更多組態供應商使用上述步驟。

Configure in Systems Manager Parameter Store

在 Systems Manager Parameter Store 中設定參數值

1. 開啟 Systems Manager 主控台。
2. 在導覽窗格中，選擇 Parameter Store (參數存放區)。
3. 建立新參數以存放在 Systems Manager 中。如需指示，請參閱《 AWS Systems Manager 使用指南》中的「建立 Systems Manager 參數 (主控台)」。
4. 複製您參數的 ARN。
5. 將以下範例政策中的 Systems Manager 許可新增至您的服務執行角色。用您的參數的 *ARN ##<ARN#####-1#123456789000###/*>。MyParameterName

```
{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "VisualEditor0",
                "Effect": "Allow",
                "Action": [
                    "ssm:GetParameterHistory",
                    "ssm:GetParametersByPath",
                    "ssm:GetParameters",
```

```
                    "ssm:GetParameter"
                ],
                "Resource": "arn:aws:ssm:us-east-1:123456789000:parameter/
MyParameterName"
            }
        ]
    }
```

6. 若要使用 Parameter Store 組態供應商，請在步驟 3 將下列程式碼行複製到工作程序組態文字方塊：

```
# 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. 如果使用 Parameter Store 組態供應商，請在步驟 5 中複製連接器組態的以下程式碼行。

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

您也可以將更多組態供應商綁定上述兩個步驟。

Configure in Amazon S3

在 Amazon S3 中設定物件/檔案

1. 開啟 Amazon S3 主控台。
2. 在 S3 中將您的物件上傳至儲存貯體。如需相關說明，請參閱上傳物件。
3. 複製您物件的 ARN。

4. 將以下範例政策中的 Amazon S3 Object Read 許可新增至您的服務執行角色。使用您物件的 ARN 來取代 *<arn:aws:s3:::MY_S3_BUCKET/path/to/custom-plugin.zip>*。

```
{
    "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. 若要使用 Amazon S3 組態供應商，請在步驟 3 中將下列程式碼行複製到工作程序組態文字方塊：

```
# define name of config provider:

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

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

6. 如果使用 Amazon S3 組態供應商，請在步驟 4 中將以下程式碼行複製至連接器組態。

```
#Example implementation for S3 object

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

您也可以將更多組態供應商綁定上述兩個步驟。

步驟 3：使用您組態供應商的資訊來建立自訂工作程序組態

1. 選取 Amazon MSK Connect 區段下的工作程序組態。

2. 選取建立工作程序組態。
3. 在工作程序組態名稱文字方塊中輸入 SourceDebeziumCustomConfig。描述為選用。
4. 根據所需的供應商複製相關組態程式碼，然後將其貼到工作程序組態文字方塊中。
5. 以下是所有三個供應商的工作程序組態範例：

```
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. 按一下「建立工作程序組態」。

步驟 4：建立連接器

1. 根據建立新連接器中的指示來建立新連接器。
2. 選擇您在 ??? 中上傳到 S3 儲存貯體的 custom-plugin.zip 檔案作為自訂外掛程式的來源。
3. 根據所需的供應商複製相關組態程式碼，然後將其貼到連接器組態欄位中。
4. 以下是所有三個供應商的連接器組態範例：

```
#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. 選取 [使用自訂組態]，並SourceDebeziumCustomConfig從 [Worker 組態] 下拉式清單中選擇
6. 依照建立連接器中指示的其餘步驟進行。

考量事項

在搭配使用 MSK 組態供應商和 Amazon MSK Connect 時，請考量以下事項：

- 在使用組態供應商時，將適當的許可指派至 IAM 服務執行角色。
- 在工作程序組態中定義組態供應商，並在連接器組態中定義其實作。
- 如果外掛程式未將這些值定義為秘密，則敏感組態值可能會顯示在連接器日誌中。Kafka Connect 會將未定義的組態值視為與任何其他純文字值相同。如需進一步了解，請參閱避免秘密顯示在連接器日誌中。
- 根據預設，在連接器使用組態供應商時，MSK Connect 會經常重新啟動連接器。若要關閉此重新啟動行為，您可以在連接器組態中將 `config.action.reload` 值設定為 `none`。

MSK Connect 的 IAM 角色和政策

主題

- 服務執行角色
- MSK Connect 的 IAM 政策範例
- 預防跨服務混淆代理人
- AWS MSK Connect 的受管理原則
- 使用 MSK Connect 的服務連結角色

服務執行角色

> **Note**
>
> Amazon MSK Connect 不支援使用服務連結角色作為服務執行角色。您必須建立個別的服務執行角色。如需如何建立自訂 IAM 角色的指示，請參閱 IAM 使用者指南中的建立角色以將許可委派給 AWS 服務。

使用 MSK Connect 建立連接器時，您必須指定要與其搭配使用的 AWS Identity and Access Management (IAM) 角色。您的服務執行角色必須具有以下信任政策，MSK Connect 才能擔任該角色。如需有關此政策中條件內容鍵的詳細資訊，請參閱 the section called “預防跨服務混淆代理人”。

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

如果要與連接器搭配使用的 Amazon MSK 叢集是使用 IAM 身分驗證的叢集，則您必須將以下許可政策新增至連接器的服務執行角色。如需有關如何尋找叢集的 UUID 以及如何建構主題 ARN 的相關資訊，請參閱 the section called “資源”。

```
{
    "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-*"
            ]
        }
    ]
}
```

視連接器類型而定，您可能還需要將允許其存取 AWS 資源的權限原則附加至服務執行角色。例如，若您的連接器需要將資料傳送至 S3 儲存貯體，則服務執行角色必須具有授予寫入該儲存貯體的許可政策。為了進行測試，您可以使用其中一個預先建立的 IAM 政策 (例如 `arn:aws:iam::aws:policy/AmazonS3FullAccess`) 來提供完整存取權。但是，為了安全起見，我們建議您使用最嚴格的原則，以允許連接器從 AWS 來源讀取或寫入接 AWS 收器。

MSK Connect 的 IAM 政策範例

若要讓非管理員使用者完整存取所有 MSK Connect 功能，請將類似以下政策的政策連接至使用者的 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"
        }
    ]
}
```

預防跨服務混淆代理人

混淆代理人問題屬於安全性問題，其中沒有執行動作許可的實體可以強制具有更多許可的實體執行該動作。在中 AWS，跨服務模擬可能會導致混淆的副問題。在某個服務 (呼叫服務) 呼叫另一個服務 (被呼叫服務) 時，可能會發生跨服務模擬。可以操縱呼叫服務來使用其許可，以其不應有存取許可的方式對其他客戶的資源採取動作。為了預防這種情況， AWS 提供的工具可協助您保護所有服務的資料，而這些服務主體已獲得您帳戶中資源的存取權。

若要限制 MSK Connect 為資源提供另一項服務的許可，我們建議在資源政策中使用 aws:SourceArn 和 aws:SourceAccount 全域條件內容鍵。如果 aws:SourceArn 值不包含帳戶 ID (例如 Amazon S3 儲存貯體 ARN 不包含帳戶 ID)，則您必須使用這兩個全域條件內容鍵來限制許可。如果同時使用這兩個全域條件內容索引鍵，且 aws:SourceArn 值包含帳戶 ID，則在相同政策陳述式中使用 aws:SourceAccount 值和 aws:SourceArn 值中的帳戶時，必須使用相同的帳戶 ID。如果您想要僅允許一個資源與跨服務存取相關聯，則請使用 aws:SourceArn。如果您想要允許該帳戶中的任何資源與跨服務使用相關聯，請使用 aws:SourceAccount。

若為 MSK Connect，aws:SourceArn 的值必須是 MSK 連接器。

防範混淆代理人問題最有效的方法，是使用 aws:SourceArn 全域條件內容金鑰，以及資源的完整 ARN。如果不知道資源的完整 ARN，或者如果您指定了多個資源，請使用 aws:SourceArn 全域條

件內容金鑰，同時使用萬用字元 (*) 表示 ARN 的未知部分。例如，*arn:aws:kafkaconnect:us-east-1:123456789012:connector/** 代表美國東部 (維吉尼亞北部) 區域中屬於帳戶 123456789012 的所有連接器。

下列範例示範如何使用 MSK Connect 中的 aws:SourceArn 和 aws:SourceAccount 全域條件內容鍵，來預防混淆代理人問題。請使用您的資訊來取代 *Account-ID* 和 *MSK-Connector-ARN*。

```
{
  "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 的受管理原則

受 AWS 管理的策略是由建立和管理的獨立策略 AWS。 AWS 受管理的策略旨在為許多常見使用案例提供權限，以便您可以開始將權限指派給使用者、群組和角色。

請記住， AWS 受管理的政策可能不會為您的特定使用案例授與最低權限權限，因為這些權限可供所有 AWS 客戶使用。我們建議您定義使用案例專屬的客戶管理政策，以便進一步減少許可。

您無法變更受 AWS 管理策略中定義的權限。如果 AWS 更新 AWS 受管理原則中定義的權限，則此更新會影響附加原則的所有主體識別 (使用者、群組和角色)。 AWS 當新的啟動或新 AWS 服務 的 API 操作可供現有服務使用時，最有可能更新 AWS 受管理策略。

如需詳細資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策。

AWS 管理策略：亞馬遜 ConnectReadOnlyAccess

此政策會授予使用者列出和描述 MSK Connect 資源所需的許可。

您可將 AmazonMSKConnectReadOnlyAccess 政策連接到 IAM 身分。

```
{
    "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 受管理的策略： KafkaConnectServiceRolePolicy

此政策會授予 MSK Connect 服務建立和管理具有標籤 `AmazonMSKConnectManaged:true` 之網路介面所需的許可。這些網路介面可讓 MSK Connect 網路存取 Amazon VPC 中的資源，例如 Apache Kafka 叢集、來源或目的地。

您無法附加 KafkaConnectServiceRolePolicy 到 IAM 實體。此政策會連接到服務連結角色，從而 MSK Connect 可代表您執行動作。

```
{
 "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 到 AWS 受管理原則的更新

檢視 MSK Connect AWS 受管理原則更新的詳細資料，因為此服務開始追蹤這些變更。

變更	描述	日期
MSK Connect 已更新唯讀政策	MSK Connect 更新了亞馬遜 MSK ConnectReadOnlyAccess 政策，以消除對上市操作的限制。	2021 年 10 月 13 日

變更	描述	日期
MSK Connect 已開始追蹤變更	MSK Connect 開始追蹤其 AWS 受管理政策的變更。	2021 年 9 月 14 日

使用 MSK Connect 的服務連結角色

Amazon MSK Connect 使用 AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是直接連結至 MSK Connect 的一種特殊 IAM 角色類型。服務連結角色由 MSK Connect 預先定義，並包含服務代表您呼叫其他 AWS 服務所需的所有權限。

服務連結角色可讓設定 MSK Connect 更為簡單，因為您不必手動新增必要的許可。MSK Connect 會定義其服務連結角色的許可，除非另有定義，否則僅有 MSK Connect 可以擔任其角色。定義的許可包括信任政策和許可政策，並且該許可政策不能連接到任何其他 IAM 實體。

如需關於支援服務連結角色的其他服務的資訊，請參閱可搭配 IAM 運作的AWS 服務，並尋找 Service-Linked Role (服務連結角色) 欄顯示為 Yes (是) 的服務。選擇具有連結的是，以檢視該服務的服務連結角色文件。

MSK Connect 的服務連結角色許可

MSK Connect 使用名為的服務連結角色 AWSServiceRoleForKafkaConnect— 允許 Amazon MSK Connect 代表您存取 Amazon 資源。

服 AWSServiceRoleForKafkaConnect 務連結角色會信任`kafkaconnect.amazonaws.com`服務擔任該角色。

如需有關角色使用之許可政策的詳細資訊，請參閱 the section called “KafkaConnectServiceRolePolicy”。

您必須設定許可，IAM 實體 (如使用者、群組或角色) 才可建立、編輯或刪除服務連結角色。如需詳細資訊，請參閱《IAM 使用者指南》中的服務連結角色許可。

建立 MSK Connect 的服務連結角色

您不需要手動建立一個服務連結角色。當您在 AWS Management Console、或 AWS API 中建立連接器時 AWS CLI，MSK Connect 會為您建立服務連結角色。

若您刪除此服務連結角色，之後需要再次建立，您可以在帳戶中使用相同程序重新建立角色。當您建立連接器時，MSK Connect 會再次為您建立服務連結角色。

編輯 MSK Connect 的服務連結角色

MSK Connect 不允許您編輯 AWSServiceRoleForKafkaConnect 服務連結的角色。因為可能有各種實體會參考服務連結角色，所以您無法在建立角色之後變更其名稱。然而，您可使用 IAM 來編輯角色描述。如需更多資訊，請參閱 IAM 使用者指南中的編輯服務連結角色。

刪除 MSK Connect 的服務連結角色

您可以使用 IAM 主控台 AWS CLI 或 AWS API 手動刪除服務連結角色。若要執行此操作，您必須先手動刪除所有 MSK Connect 連接器，然後再手動刪除該角色。如需詳細資訊，請參閱《IAM 使用者指南》中的刪除服務連結角色。

MSK Connect 服務連結角色支援的區域

MSK Connect 支援在所有提供此服務的區域中使用服務連結角色。如需詳細資訊，請參閱 AWS 區域與端點。

啟用 Amazon MSK Connect 的網際網路存取

如果您的 Amazon MSK Connect 連接器需要存取網際網路，建議您使用下列 Amazon Virtual Private Cloud (VPC) 設定來啟用該存取權。

- 使用私有子網路來設定連接器。
- 在公有子網路中為您的 VPC 建立公有 NAT 閘道或 NAT 執行個體。如需詳細資訊，請參閱《Amazon Virtual Private Cloud使用指南》中的使用 NAT 裝置將子網路連線到網際網路或其他 VPC 頁面。
- 允許流量從私有子網路輸出至 NAT 閘道或執行個體。

設定 Amazon MSK Connect 的 NAT 閘道

以下步驟會顯示設定 NAT 閘道以啟用連接器網際網路存取權的方法。您必須先完成這些步驟，才能在私有子網路中建立連接器。

先決條件

請確認您已具備以下物件。

- 與叢集相關聯的 Amazon Virtual Private Cloud (VPC) 識別碼。例如，vpc-123456ab。

- VPC 中私有子網路的 ID。例如，subnet-a1b2c3de、subnet-f4g5h6ij 等。您必須使用私有子網路來設定連接器。

啟用連接器的網際網路存取

1. 請在以下位置開啟 Amazon Virtual Private Cloud 主控台。 https://console.aws.amazon.com/vpc/
2. 使用描述性名稱為 NAT 閘道建立公有子網路，並記下子網路 ID。如需詳細說明，請參閱在 VPC 中建立子網路。
3. 建立網際網路閘道，讓您的 VPC 可和網際網路通訊，並記下閘道 ID。將網際網路閘道連接至 VPC。如需說明，請參閱建立並連接網際網路閘道。
4. 佈建公有 NAT 閘道，以便讓私有子網路中的主機可連線到您的公有子網路。在建立 NAT 閘道時，請選取您稍早建立的公有子網路。如需說明，請參閱建立 NAT 閘道。
5. 設定路由表。您總共須有兩個路由表才能完成此設定。您應已經擁有與 VPC 同時自動建立的主路由表。在此步驟中，您可為公有子網路建立額外的路由表。

   a. 使用以下設定來修改 VPC 的主路由表，以便讓私有子網路將流量路由到 NAT 閘道。如需說明，請參閱《Amazon Virtual Private Cloud使用者指南》中的使用路由表。

      私有 MSKC 路由表

屬性	Value
Name tag (名稱標籤)	我們建議您為此路由表提供描述性名稱標籤，有助於您識別。例如，Private MSKC。
相關聯的子網路	您的私有子網路
為 MSK Connect 啟用網際網路存取權的路由	• 目的地：0.0.0.0/0 • 目標：您的 NAT 閘道 ID。例如，nat-12a345bc6789efg1h。
適用於內部流量的本機路由	• 目的地：10.0.0.0/16。此值可能會依 VPC 的 CIDR 區塊而略有差異。 • 目標：本機

   b. 依照建立自訂路由表中的說明，建立公有子網路的路由表。在建立該表時，請在名稱標籤欄位中輸入描述性名稱，以協助您識別與該表相關聯的子網路。例如，Public MSKC。

c. 使用以下設定來設定您的 Public MSKC 路由表。

屬性	Value
Name tag (名稱標籤)	Public MSKC 或您所選的其他描述性名稱
相關聯的子網路	您具有 NAT 閘道的公有子網路
為 MSK Connect 啟用網際網路存取權的路由	• 目的地：0.0.0.0/0 • 目標：您的網際網路閘道 ID。例如，igw-1a234bc5。
適用於內部流量的本機路由	• 目的地：10.0.0.0/16。此值可能會依 VPC 的 CIDR 區塊而略有差異。 • 目標：本機

私有 DNS 主機名稱

藉由 MSK Connect 中的私有 DNS 主機名稱支援，您可以設定連接器來參照公有或私有網域名稱。支援會依據 VPC DHCP 選項組中指定的 DNS 伺服器而定。

DHCP 選項集是 EC2 執行個體在 VPC 中使用的一組網路組態，以透過虛擬網路進行通訊。每個 VPC 都具有預設 DHCP 選項集，但如果您要讓 VPC 中的執行個體使用不同的 DNS 伺服器 (而不是 Amazon 提供之 DNS 伺服器) 來進行網域名稱解析，則可以建立自訂 DHCP 選項集。請參閱 Amazon VPC 中的 DHCP 選項集。

在使用 MSK Connect 中納入私有 DNS 解析能力/功能之前，連接器會使用服務 VPC DNS 解析器來進行客戶連接器的 DNS 查詢。連接器並未使用客戶 VPC DHCP 選項集中定義的 DNS 伺服器進行 DNS 解析。

連接器僅能參考客戶連接器組態或外掛程式中可公開解析的主機名稱。它們無法解析在私有託管區域中定義的私有主機名稱，或在其他客戶網路中使用 DNS 伺服器。

若無私有 DNS，則選擇讓資料庫、資料倉儲和系統 (例如自身無法存取網際網路的 VPC 中的 Secrets Manager) 的客戶，將無法使用 MSK 連接器。客戶經常會使用私有 DNS 主機名稱來符合企業安全性狀態。

主題

為您的連接器設定 VPC DHCP 選項集

連接器建立後，連接器會自動使用其 VPC DHCP 選項集中定義的 DNS 伺服器。在建立連接器之前，請確認已針對連接器的 DNS 主機名稱解析需求設定 VPC DHCP 選項集。

您在 MSK Connect 提供私有 DNS 主機名稱功能之前所建立的連接器，會繼續使用先前的 DNS 解析組態，且無需進行任何修改。

若您僅需連接器中的可公開解析 DNS 主機名稱解析，為了能更輕鬆地進行設定，建議您在建立連接器時使用帳戶的預設 VPC。如需有關 Amazon 提供之 DNS 伺服器或 Amazon Route 53 Resolver 的詳細資訊，請參閱《Amazon VPC 使用者指南》中的 Amazon DNS 伺服器。

若您需要解析私有 DNS 主機名稱，請確認連接器建立期間傳送的 VPC 已正確設定其 DHCP 選項。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的使用 DHCP 選項集。

在您為私有 DNS 主機名稱解析設定 DHCP 選項集時，請確認連接器可連接至您在 DHCP 選項集中設定的自訂 DNS 伺服器。否則，您的連接器建立會失敗。

在您自訂 VPC DHCP 選項集後，隨後在該 VPC 中建立的連接器會使用您在選項集中指定的 DNS 伺服器。若您在建立連接器後變更選項集，則連接器會在幾分鐘後採用新選項集中的設定。

VPC 的 DNS 屬性

請確認您已正確設定 VPC DNS 屬性，如《Amazon VPC 使用者指南》中的 VPC 的 DNS 屬性和 DNS 主機名稱所述。

請參閱《Amazon Route 53 開發人員指南》中的在 VPC 和您網路之間解析 DNS 查詢，了解有關使用傳入和傳出解析器端點將其他網路連接至 VPC 以與連接器搭配使用的資訊。

失敗處理

本節會描述可能發生的與 DNS 解析相關聯的連接器建立失敗，以及解決問題的建議行動。

失敗	建議動作
若 DNS 解析查詢失敗，或者無法連接器無法連線 DNS 伺服器，則連接器建立會失敗。	如果您已為連接器設定這些記錄檔，您可以在 CloudWatch 記錄檔中看到因 DNS 解析查詢失敗而導致的連接器建立失敗。 檢查 DNS 伺服器組態，並確保連接器與 DNS 伺服器之間的網路連線。
若在連接器執行時變更 VPC DHCP 選項集中的 DNS 伺服器組態，則來自連接器的 DNS 解析查詢可能會失敗。若 DNS 解析失敗，部分連接器任務可能會進入失敗狀態。	如果您已為連接器設定這些記錄檔，您可以在 CloudWatch 記錄檔中看到因 DNS 解析查詢失敗而導致的連接器建立失敗。 失敗的任務應會自動重新啟動，以恢復連接器。若未發生此情況，您可以聯絡支援團隊以重新啟動連接器的失敗任務，或者您可重新建立連接器。

MSK Connect 的日誌

MSK Connect 能寫入日誌事件，您可用來對連接器進行偵錯。在建立連接器時，您可以指定下列零個或多個日誌目的地：

- Amazon CloudWatch 日誌：您可以指定要 MSK Connect 傳送連接器日誌事件的日誌群組。如需如何建立記錄群組的詳細資訊，請參閱CloudWatch 記錄檔使用手冊中的建立記錄群組。
- Amazon S3：您可以指定要 MSK Connect 向其傳送連接器日誌事件的 S3 儲存貯體。如需有關建立 S3 儲存貯體的詳細資訊，請參閱《Amazon S3 使用者指南》中的建立儲存貯體。
- Amazon 資料 Firehose：您可以指定要 MSK Connect 傳送連接器日誌事件的交付串流。如需如何建立交付串流的詳細資訊，請參閱 Firehose 使用者指南中的建立 Amazon 資料 Firehose 交付串流。

若要進一步了解如何設定日誌，請參閱《Amazon CloudWatch Logs 使用者指南》中的啟用從 AWS 服務記錄日誌。

MSK Connect 會發出下列類型的日誌事件：

Level	描述
INFO	啟動和關閉時需要關注的執行期事件。
WARN	執行期狀況並非錯誤，但不想要或未預期該執行期狀況。
FATAL	導致過早終止的嚴重錯誤。
ERROR	不嚴重的意外狀況和執行期錯誤。

以下是傳送至 CloudWatch 記錄檔的記錄事件範例：

```
[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)
```

避免秘密顯示在連接器日誌中

> **Note**
>
> 如果外掛程式未將這些值定義為秘密，則敏感組態值可能會顯示在連接器日誌中。Kafka Connect 會將未定義的組態值視為與任何其他純文字值相同。

如果您的外掛程式將屬性定義為秘密，則 Kafka Connect 會在連接器日誌中遮蔽該屬性的值。例如，下列連接器日誌會示範若外掛程式將 `aws.secret.key` 定義為 PASSWORD 類型，則其值將被取代為 **`[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)
```

若要避免秘密出現在連接器日誌檔案中，外掛程式開發人員必須使用 Kafka Connect 列舉常數 `ConfigDef.Type.PASSWORD` 來定義敏感屬性。當屬性是 `ConfigDef.Type.PASSWORD` 類型時，即使該值以純文字形式傳送，Kafka Connect 也會從連接器日誌中排除其值。

監控 MSK Connect

監控是維持 MSK Connect 和其他 AWS 解決方案的可靠性、可用性和效能的重要組成部分。Amazon 會即時 CloudWatch 監控您的 AWS 資源和執行 AWS 的應用程式。您可以收集和追蹤指標、建立自訂儀板表，以及設定警示，在特定指標達到您指定的閾值時通知您或採取動作。例如，您可以 CloudWatch 追蹤連接器的 CPU 使用率或其他指標，以便在需要時增加其容量。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。

下表顯示 MSK Connect 傳送到`ConnectorName`維度 CloudWatch 底下的測量結果。MSK Connect 預設會提供這些指標，而且無需額外付費。 CloudWatch 將這些指標保留 15 個月，以便您可以存取歷史資訊，並更好地瞭解連接器的效能。您也可以設定留意特定閾值的警示，當滿足這些閾值時傳送通知或採取動作。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。

MSK Connect 指標

指標名稱	描述
`BytesInPerSec`	連接器接收的總位元組數。
`BytesOutPerSec`	連接器傳送的總位元組數。
`CpuUtilization`	CPU 使用率 (依系統和使用者)。
`ErroredTaskCount`	錯誤的任務數。
`MemoryUtilization`	工作程序執行個體的總記憶體百分比，而非僅是目前使用中的 Java 虛擬機器 (JVM) 堆積記憶體。JVM 通常不會釋放記憶體使其回到作業系統。因此，JVM 堆大小（MemoryUtilization）通常以最小堆大小開始，該堆大小逐漸增加到約 80-90% 的穩定最大值。JVM 堆積使用量可能

指標名稱	描述
	會隨連接器的實際記憶體使用量變更而增加或減少。
RebalanceCompletedTotal	此連接器完成的重新平衡總數。
RebalanceTimeAvg	連接器重新平衡所花費的平均時間 (毫秒)。
RebalanceTimeMax	連接器重新平衡所花費的最大時間 (毫秒)。
RebalanceTimeSinceLast	自此連接器完成最近重新平衡之後的時間 (毫秒)。
RunningTaskCount	連接器中執行的任務數。
SinkRecordReadRate	從 Apache Kafka 或 Amazon MSK 叢集平均每秒讀取的記錄數。
SinkRecordSendRate	平均每秒從轉換輸出並傳送至目的地的記錄數。此數字不包含已篩選的記錄。
SourceRecordPollRate	平均每秒產生或輪詢的記錄數。
SourceRecordWriteRate	平均每秒從輸出轉換並寫入 Apache Kafka 或 Amazon MSK 叢集的記錄數。
TaskStartupAttemptsTotal	連接器嘗試啟動的任務總數。您可以使用此指標來識別任務啟動嘗試中的異常情況。
TaskStartupSuccessPercentage	連接器成功啟動任務的平均百分比。您可以使用此指標來識別任務啟動嘗試中的異常情況。
WorkerCount	連接器中執行的工作程序數目。

範例

本節包含的範例有助您設定 Amazon MSK Connect 資源，例如常見的第三方連接器和組態供應商。

主題

- Amazon S3 目的地連接器
- 具有組態供應商的 Debezium 來源連接器

Amazon S3 目的地連接器

此範例顯示如何使用匯流 Amazon S3 接收器 Connec t 器， AWS CLI 以及在 MSK 連接中建立 Amazon S3 接收器連接器。

1. 複製以下 JSON 並貼到新檔案中。使用對應至 Amazon MSK 叢集的引導程序伺服器連線字串以及叢集的子網路和安全群組 ID 的值來取代預留位置字串。如需有關如何設定服務執行角色的詳細資訊，請參閱 the section called “(IAM) 角色和政策”。

```
{
    "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. 在上一個步 AWS CLI 驟中儲存 JSON 檔案的資料夾中執行下列命令。

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

以下是成功執行命令時所得到的輸出範例。

```
{
    "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"
}
```

具有組態供應商的 Debezium 來源連接器

此範例顯示如何使用具有與 MySQL 相容 Amazon Aurora 資料庫的 Debezium MySQL 連接器外掛程式做為來源。在此範例中，我們也會設定開放原始碼 AWS Secrets Manager Config Provider，以在 AWS Secrets Manager中將資料庫憑證外部化。如需有關組態供應商的詳細資訊，請參閱 使用組態供應商來外部化敏感資訊。

> **Important**
>
> Debezium MySQL 連接器外掛程式僅支援一個任務，不適用於 Amazon MSK Connect 的自動擴展容量模式。您應改用佈建容量模式，並在連接器組態中將 `workerCount` 設定為等於一。如需有關 MSK Connect 容量模式的詳細資訊，請參閱 連接器容量。

開始之前

您的連接器必須能夠存取網際網路，以便它可以與您以 AWS Secrets Manager 外的服務互動 Amazon Virtual Private Cloud。本節中的步驟有助您完成以下任務，以啟用網際網路存取。

- 設定託管 NAT 閘道的公有子網路，並將流量路由至 VPC 中的網際網路閘道。
- 建立預設路由，將您的私有子網路流量導向 NAT 閘道。

如需詳細資訊，請參閱 啟用 Amazon MSK Connect 的網際網路存取。

先決條件

您需先準備好以下事項，才可啟用網際網路存取：

- 與叢集相關聯的 Amazon Virtual Private Cloud (VPC) 識別碼。例如，vpc-123456ab。
- VPC 中私有子網路的 ID。例如，subnet-a1b2c3de、subnet-f4g5h6ij 等。您必須使用私有子網路來設定連接器。

啟用連接器的網際網路存取

1. 請在以下位置開啟 Amazon Virtual Private Cloud 主控台。 https://console.aws.amazon.com/vpc/
2. 使用描述性名稱為 NAT 閘道建立公有子網路，並記下子網路 ID。如需詳細說明，請參閱在 VPC 中建立子網路。
3. 建立網際網路閘道，讓您的 VPC 可和網際網路通訊，並記下閘道 ID。將網際網路閘道連接至 VPC。如需說明，請參閱建立並連接網際網路閘道。
4. 佈建公有 NAT 閘道，以便讓私有子網路中的主機可連線到您的公有子網路。在建立 NAT 閘道時，請選取您稍早建立的公有子網路。如需說明，請參閱建立 NAT 閘道。
5. 設定路由表。您總共須有兩個路由表才能完成此設定。您應已經擁有與 VPC 同時自動建立的主路由表。在此步驟中，您可為公有子網路建立額外的路由表。

a. 使用以下設定來修改 VPC 的主路由表，以便讓私有子網路將流量路由到 NAT 閘道。如需說明，請參閱《Amazon Virtual Private Cloud使用者指南》中的使用路由表。

   私有 MSKC 路由表

屬性	Value
Name tag (名稱標籤)	我們建議您為此路由表提供描述性名稱標籤，有助於您識別。例如，Private MSKC。
相關聯的子網路	您的私有子網路
為 MSK Connect 啟用網際網路存取權的路由	• 目的地：0.0.0.0/0 • 目標：您的 NAT 閘道 ID。例如，nat-12a345bc6789efg1h。
適用於內部流量的本機路由	• 目的地：10.0.0.0/16。此值可能會依 VPC 的 CIDR 區塊而略有差異。 • 目標：本機

b. 依照建立自訂路由表中的說明，建立公有子網路的路由表。在建立該表時，請在名稱標籤欄位中輸入描述性名稱，以協助您識別與該表相關聯的子網路。例如，Public MSKC。

c. 使用以下設定來設定您的 Public MSKC 路由表。

屬性	Value
Name tag (名稱標籤)	Public MSKC 或您所選的其他描述性名稱
相關聯的子網路	您具有 NAT 閘道的公有子網路
為 MSK Connect 啟用網際網路存取權的路由	• 目的地：0.0.0.0/0 • 目標：您的網際網路閘道 ID。例如，igw-1a234bc5。
適用於內部流量的本機路由	• 目的地：10.0.0.0/16。此值可能會依 VPC 的 CIDR 區塊而略有差異。 • 目標：本機

現在已為 Amazon MSK Connect 啟用網際網路存取，您可以建立連接器了。

建立 Debezium 來源連接器

1. 建立自訂外掛程式

   a. 從 Debezium 網站下載 MySQL 連線器外掛程式的最新穩定版本。請記下您下載的 Debezium 發行版本 (版本 2.x 或較舊的 1.x 系列)。稍後在此程序中，您將根據您的 Debezium 版本建立連接器。

   b. 下載並解壓縮 AWS Secrets Manager Config Provider。

   c. 將以下封存放入相同的目錄中：

   - debezium-connector-mysql 資料夾
   - jcusten-border-kafka-config-provider-aws-0.1.1 資料夾

   d. 將您在上一個步驟中建立的目錄壓縮為 ZIP 檔案，然後將該 ZIP 檔案上傳至 S3 儲存貯體。如需說明，請參閱《Amazon S3 使用者指南》中的上傳物件。

   e. 複製以下 JSON 並貼到檔案中。例如 debezium-source-custom-plugin.json。將 *< example-custom-plugin-name >* 取代為您希望外掛程式具有的名稱，*< arn-of-your-s 3 ##### >* 取代為您上傳 ZIP 檔案的 S3 儲存貯體的 ARN，以及*<file-key-of-ZIP-object>*您上傳至 S3 之 ZIP 物件的檔案金鑰。

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

   f. 從儲存 JSON 檔案的資料夾執行下列 AWS CLI 命令，以建立外掛程式。

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

   您應該會看到類似以下範例的輸出。

```
{
    "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. 執行以下命令來檢查外掛程式狀態。狀態應從 CREATING 變更為 ACTIVE。使用您在上一個命令輸出中獲得的 ARN 來取代 ARN 預留位置。

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

2. 設定 AWS Secrets Manager 和建立資料庫認證的密碼

a. 前往以下位置開啟機密管理員控制台：https://console.aws.amazon.com/secretsmanager/。

b. 建立新秘密以存放您的資料庫登入憑證。如需說明，請參閱《AWS Secrets Manager使用者指南》中的建立秘密。

c. 複製您秘密的 ARN。

d. 將以下範例政策中的 Secrets Manager 許可新增至您的 服務執行角色。替換 *<arn#AW############-1#123456789000####-* 1234> 與您的秘密的 ARN。MySecret

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

如需有關如何新增 IAM 許可的說明，請參閱《IAM 使用者指南》中的新增和移除 IAM 身分許可。

3. 使用您組態供應商的資訊來建立自訂工作程序組態

   a. 將以下工作程序組態屬性複製到檔案中，並使用對應至您案例的值來取代預留位置字串。若要進一步了解 AWS Secrets Manager 組態提供者的組態屬性，請參閱外掛程SecretsManagerConfigProvider式的說明文件中的。

```
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. 執行下列 AWS CLI 命令以建立您的自訂 Worker 組態。

      取代以下的值：

      - *< my-worker-config-name >*-自訂 Worker 組態的描述性名稱
      - *< encoded-properties-file-content-string>-#*在上一個步驟中複製的純文字屬性的 base64 編碼版本

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

4. 建立連接器

   a. 複製以下對應至您 Debezium 版本 (2.x 或 1.x) 的 JSON，然後將其貼至新檔案中。使用對應至您案例的值來取代 *<placeholder>* 字串。如需有關如何設定服務執行角色的詳細資訊，請參閱 the section called “(IAM) 角色和政策”。

      請注意，組態會使用類似 `${secretManager:MySecret-1234:dbusername}` 的變數 (而非純文字) 來指定資料庫憑證。使用您秘密的名稱來取代 *MySecret-1234*，然後加入您要擷取的索引鍵名稱。您也必須使用自訂工作程序組態的 ARN 來取代 *<arn-of-config-provider-worker-configuration>*。

Debezium 2.x

針對 Debezium 2.x 版本，請複製以下 JSON 並將其貼至新檔案中。使用對應至您案例的值來取代 *<placeholder>* 字串。

```
{
 "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

針對 Debezium 1.x 版本，請複製以下 JSON 並將其貼至新檔案中。使用對應至您案例的值來取代 *<placeholder>* 字串。

```
{
  "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. 在上一個步 AWS CLI 驟中儲存 JSON 檔案的資料夾中執行下列命令。

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

以下是成功執行命令時所得到的輸出範例。

```
{
    "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"
}
```

如需具有詳細步驟的 Debezium 連接器範例，請參閱 Introducing Amazon MSK Connect - Stream Data to and from Your Apache Kafka Clusters Using Managed Connectors。

最佳實務

以此作為參考資訊，快速找到使用 Amazon MSK Connect 發揮最大效能的建議。

主題

從連接器連線

以下最佳實務可改善您與 Amazon MSK Connect 的連線效能。

請勿重疊 Amazon VPC 對等互連或 Transit Gateway 的 IP

若您使用 Amazon VPC 對等互連或 Transit Gateway 搭配 Amazon MSK Connect，請勿將連接器設定為透過 CIDR 範圍中的 IP 連線對等互聯的 VPC 資源：

- "10.99.0.0/16"
- "192.168.0.0/16"
- "172.21.0.0/16"

Amazon MSK Connect 遷移指南

本節說明如何將 Apache 卡夫卡 Connect 器應用程式遷移到 Amazon 的 Apache 卡夫卡 Connect 器管理串流 (Amazon MSK 連接)。

主題

使用 Amazon MSK Connect 的好處

Apache Kafka 是用於擷取和處理即時資料串流的最廣泛採用的開放原始碼串流平台之一。使用 Apache Kafka，您可以分離和獨立擴展資料產生和資料消耗的應用程式。

卡夫卡 Connect 是構建和運行流媒體應用程序與 Apache 卡夫卡的一個重要組成部分。卡夫卡 Connect 提供了卡夫卡和外部系統之間移動數據的標準化方式。Kafka Connect 具有高度可擴展性，並

且可以處理大量數據 Kafka Connect 提供了一套功能強大的 API 操作和工具，用於配置，部署和監視連接器，這些連接器可在卡夫卡主題和外部系統之間移動數據。您可以使用這些工具自訂和擴充 Kafka Connect 的功能，以滿足串流應用程式的特定需求。

當您自行操作 Apache 卡夫卡 Connect 叢集時，或嘗試將開放原始碼 Apache Kafka Connect 應用程式遷移到. AWS這些挑戰包括設定基礎架構和部署應用程式所需的時間、設定自我管理的 Apache Kafka Connect 叢集時的工程障礙，以及管理營運開銷。

為了解決這些挑戰，我們建議您使用 Amazon 受管串流的 Apache 卡夫卡 Connect（Amazon MSK Connect）將您的開放原始碼 Apache 卡夫卡 Connect 應用程式遷移到。 AWS Amazon MSK Connect 可簡化使用 Kafka Connect 在 Apache Kafka 叢集和外部系統 (例如資料庫、搜尋索引和檔案系統) 之間進行資料串流和串流資料的流程。

以下是遷移到 Amazon MSK Connect 的一些好處：

- 消除營運開銷 — Amazon MSK Connect 消除了與修補、佈建和擴展 Apache Kafka Connect 叢集相關的操作負擔。Amazon MSK Connect 會持續監控 Connect 叢集的運作狀態，並自動執行修補和版本升級，而不會對您的工作負載造成任何干擾。
- 自動重新啟動 Connect 任務 — Amazon MSK Connect 可以自動復原失敗的任務，以減少生產中斷情況。工作失敗可能是因為暫時性錯誤所造成，例如違反 Kafka 的 TCP 連線限制，以及當新 Worker 加入接收器的用戶群組時，工作重新平衡。
- 自動水平和垂直擴展 — Amazon MSK Connect 可讓連接器應用程式自動擴展，以支援更高的輸送量。Amazon MSK Connect 會為您管理擴展。您只需要指定 auto 擴展群組中的 Worker 數量和使用率閾值即可。您可以使用 Amazon MSK Connect `UpdateConnector` API 操作在 1 到 8 個 vCPUs 之間垂直擴展或縮減 vCPUs，以支援可變輸送量。
- 私有網路連線能力 — Amazon MSK Connect 透過使用私有 DNS 名稱私有連線到來源 AWS PrivateLink 和接收系統。

遷移到 Amazon MSK Connect

本節簡要說明卡夫卡 Connect 和 Amazon MSK Connect 所使用的狀態管理主題。本節也涵蓋移轉來源和接收器連接器的程序。

主題

- 卡夫卡 Connect 使用的內部主題
- Amazon MSK Connect 應用程式的狀態管理
- 將來源 Connect 器遷移到 Amazon MSK 連接

- 將接收器 Connect 器遷移到 Amazon MSK 連接

卡夫卡 Connect 使用的內部主題

在分散式模式下執行的 Apache Kafka Connect 應用程式會使用 Kafka 叢集和群組成員資格中的內部主題來儲存其狀態。下列是與 Kafka Connect 應用程式所使用的內部主題相對應的組態值：

- 組態主題, 透過指定 `config.storage.topic`

  在組態主題中，Kafka Connect 會儲存使用者已啟動的所有連接器和工作的組態。每次使用者更新連接器的組態，或當連接器要求重新設定時 (例如，連接器偵測到它可以啟動更多工作) 時，就會向此主題發出記錄。本主題已啟用壓縮，因此它始終保留每個實體的最後一個狀態。

- 偏移主題, 透過指定 `offset.storage.topic`

  在偏移主題中，卡夫卡 Connect 存儲源連接器的偏移量。與組態主題一樣，偏移主題已啟用壓縮。本主題僅用於將資料從外部系統產生資料的來源連接器寫入來源位置。接收器連接器，從卡夫卡讀取數據並發送到外部系統，通過使用常規的卡夫卡消費者組存儲其消費者偏移量。

- 狀態主題, 透過指定 `status.storage.topic`

  在狀態主題中，卡夫卡 Connect 存儲連接器和任務的當前狀態。本主題用作 REST API 使用者查詢之資料的中心位置。本主題允許使用者查詢任何 Worker，但仍可取得所有執行中外掛程式的狀態。與組態和偏移主題一樣，狀態主題也會啟用壓實。

除了這些主題，卡夫卡 Connect 還廣泛使用了卡夫卡的組成員資格 API。群組會以連接器名稱命名。例如，對於名為 file-接收器的連接器，該群組被命名 connect-file-sink為。群組中的每個用戶都會為單一工作提供記錄。您可以使用一般用戶群組工具來擷取這些群組及其位移，例如`Kafka-consumer-group.sh`。對於每個接收器連接器，Connect 執行階段會執行從卡夫卡擷取記錄的一般用戶群組。

Amazon MSK Connect 應用程式的狀態管理

根據預設，Amazon MSK Connect 會在卡夫卡叢集中為每個 Amazon MSK 連接器建立三個單獨的主題，以存放連接器的組態、偏移和狀態。預設主題名稱的結構如下：

- *`###-## _ ###-## _ ### ID`*
- *`__msk_#####_#####_######`*
- *`__msk_ ### _ ## _ ###-## _ ### ID`*

> **Note**
>
> 若要提供來源連接器之間的位移連續性，您可以使用您選擇的偏移儲存主題，而不是預設主題。指定偏移儲存主題有助您完成任務，例如，建立來源連接器從上一個連接器的最後一個偏移恢復讀取。若要指定偏移儲存主題，請在建立連接器之前，在 Amazon MSK Connect 工作者組態中提供offset.storage.topic屬性值。

將來源 Connect 器遷移到 Amazon MSK 連接

源 Connect 器是從外部系統導入記錄到卡夫卡阿帕奇卡夫卡連接應用程序。本節說明將執行現場部署的 Apache Kafka Connect 來源連接器應用程式移轉到 Amazon MSK Connect 上 AWS 執行的自我管理 Kafka Connect 叢集的程序。

Kafka Connect 來源連接器應用程式會將偏移儲存在以設定組態屬性設定的值命名的主題中。`offset.storage.topic`以下是 JDBC 連接器的範例偏移訊息，此 JDBC 連接器正在執行兩個工作，這些工作會從兩個名為`movies`和`shows`的不同表格匯入資料。從表格影片匯入的最近一列具有的主要識別碼`18343`。從顯示表格匯入的最新資料列具有的主要識別碼`732`。

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

若要將來源 Connect 器遷移到 Amazon MSK 連接，請執行下列動作：

1. 從現場部署或自我管理的 Kafka Connect 叢集提取連接器程式庫，以建立 Amazon MSK Connect 自訂外掛程式。
2. 建立 Amazon MSK Connect 工作者屬性 `key.converter`、`value.converter`，並將屬性和`offset.storage.topic`設定為在現有 Kafka Connect 叢集中執行的 Kafka 連接器所設定的相同值。
3. 在現有的 Kafka Connect 叢集上發出`PUT /connectors/`*`connector-name`*`/pause`要求，暫停現有叢集上的連接器應用程式。
4. 確定連接器應用程式的所有工作都已完全停止。您可以在現有的 Kafka Connect 叢集上發出`GET /connectors/`*`connector-name`*`/status`要求，或使用針對屬性設定的主題名稱中的訊息來停止工作。`status.storage.topic`
5. 從現有叢集取得連接器組態。您可以透過在現有叢集上`GET /connectors/`*`connector-name`*`/config/`提出要求，或使用針對內容設定的主題名稱中的訊息來取得連接器組態`config.storage.topic`。

6. 使用與現有叢集相同的名稱建立新的 Amazon MSK 連接器。使用您在步驟 1 中建立的連接器自訂外掛程式、您在步驟 2 中建立的 Worker 屬性，以及您在步驟 5 中萃取的連接器組態來建立此連接器。
7. 當 Amazon MSK 連接器狀態為時`active`，請檢視記錄以確認連接器已開始從來源系統匯入資料。
8. 透過提出`DELETE /connectors/`*`connector-name`*要求刪除現有叢集中的連接器。

將接收器 Connect 器遷移到 Amazon MSK 連接

接收器 Connect 器是 Apache 卡夫卡連接從卡夫卡數據導出到外部系統的應用程序。本節說明將執行現場部署或自我管理的 Kafka Connect 叢集執行至 Amazon MSK Connect 叢集的 Apache Kafka Connect 器接收器連接器應用程式的程序。 AWS

卡夫卡 Connect 接器接收器使用卡夫卡組成員 API 和存儲偏移量在相同的`__consumer_offset`主題作為一個典型的消費者應用程序。此行為簡化了接收器連接器從自我管理叢集遷移到 Amazon MSK Connect 的過程。

若要將接收器 Connect 器遷移至 Amazon MSK 連接器，請執行下列動作：

1. 從現場部署或自我管理的 Kafka Connect 叢集提取連接器程式庫，以建立 Amazon MSK Connect 自訂外掛程式。
2. 建立 Amazon MSK Connect 工作者屬性，並將屬性`key.converter`和`value.converter`設定為在現有 Kafka Connect 叢集中執行的 Kafka 連接器所設定的相同值。
3. 在現有的 Kafka Connect 叢集上發出`PUT /connectors/`*`connector-name`*`/pause`要求，暫停現有叢集上的連接器應用程式。
4. 確定連接器應用程式的所有工作都已完全停止。您可以在現有的 Kafka Connect 叢集上發出`GET /connectors/`*`connector-name`*`/status`要求，或使用針對屬性設定的主題名稱中的訊息來停止工作。`status.storage.topic`
5. 從現有叢集取得連接器組態。您可以透過在現有叢集上發出`GET /connectors/`*`connector-name`*`/config`要求，或使用針對內容設定的主題名稱中的訊息來取得連接器組態`config.storage.topic`。
6. 建立與現有叢集相同名稱的新 Amazon MSK 連接器。使用您在步驟 1 中建立的連接器自訂外掛程式、您在步驟 2 中建立的 Worker 屬性，以及您在步驟 5 中萃取的連接器組態來建立此連接器。
7. 當 Amazon MSK 連接器狀態為時`active`，請檢視記錄以確認連接器已開始從來源系統匯入資料。
8. 透過提出`DELETE /connectors/`*`connector-name`*要求刪除現有叢集中的連接器。

疑難排解 Amazon MSK Connect

下列資訊有助您針對使用 MSK Connect 時可能發生的問題，進行疑難排解。您也可以將問題張貼到 AWS re:Post。

連接器無法存取在公用網際網路上託管的資源

請參閱啟用 Amazon MSK Connect 的網際網路存取。

連接器的執行中任務數量不等於 tasks.max 中指定的任務數量

以下是連接器使用的任務數量少於 tasks.max 組態所指定數量的一些原因：

- 部分連接器實作會限制可用的任務數量。例如，MySQL 的 Debezium 連接器僅限使用單一任務。
- 在使用自動調整規模容量模式時，Amazon MSK Connect 會以與連接器中執行的工作程序數量和每個工作程序的 MCU 數量成比例的值，來覆寫連接器的 tasks.max 屬性。
- 針對目的地連接器，平行處理數量 (任務數量) 不得超過主題分區的數量。雖然您可以將 tasks.max 設定為大於該值，但單一分區一次不會被多個任務處理。
- 在 Kafka Connect 2.7.x 中，預設的取用者分區指派者是 `RangeAssignor`。此指派者的行為是將每個主題的第一個分區提供給單一取用者、將每個主題的第二個分區提供給單一取用者等。這表示使用 `RangeAssignor` 之目的地連接器的作用中任務數量上限會等於任何單一主題中所取用的分區數量上限。若這不適用於您的使用案例，則您應該建立工作程序組態，並在其中將 `consumer.partition.assignment.strategy` 屬性設定為更適合的取用者分區指派者。請參閱卡夫卡 2.7 接口 ConsumerPartitionAssignor：所有已知的實現類。

MSK Replicator

什麼是 Amazon MSK Replicator？

Amazon MSK 複寫器是 Amazon MSK 複寫器的一項功能，可讓您在不同或相同 AWS 區域的 Amazon MSK 叢集之間可靠地複寫資料。使用 MSK Replicator，您可以輕鬆建置具備區域彈性的串流應用程式，以提高可用性和業務連續性。MSK Replicator 提供跨 MSK 叢集的自動非同步複寫功能，無需撰寫自訂程式碼、管理基礎設施或設定跨區域聯網。

MSK Replicator 會自動擴充基礎資源，讓您可以隨需複寫資料，而無需監控或擴展容量。MSK Replicator 也會複寫必要的 Kafka 中繼資料，包括主題設定、存取控制清單 (ACL) 和取用者群組偏移。如果某個區域發生未預期的事件，您可以容錯移轉至其他 AWS 區域，並順暢地繼續處理。

MSK Replicator 支援跨區域複寫 (CRR) 和相同區域複寫 (SRR)。在跨區域複寫中，來源和目標 MSK 叢集位於不同 AWS 的區域。在相同區域複寫中，來源和目標 MSK 叢集都位於相同 AWS 區域中。您必須先建立來源和目標 MSK 叢集，才能將它們與 MSK Replicator 搭配使用。

> **Note**
>
> MSK 複製器支援下列 AWS 區域：美國東部 (us-east-1、維吉尼亞北部)、美國東部 (us-east-2、俄亥俄)、美國西部 (美國西部 us-west-2、奧勒岡)、歐洲 (歐洲 eu-west-1、愛爾蘭)、歐洲 (歐洲-中央 -1、法蘭克福)、亞太區域 (ap-southeast-1、新加坡)；亞太區域 (北歐)、雪梨 ap-southeast-2 eu-north-1、斯德哥爾摩)、亞太區域 (ap-south-1 1、孟買)、歐洲 (歐洲至西 3、巴黎)、南美洲 (sa-east-1、聖保羅)、亞太區域 (AP-東北 -2、首爾)、歐洲 (歐洲-eu-west-2 部 -2、倫敦)、亞太區域 (東北 1、東京)、美國西部 1、美國西部 1、美國西部 1、美國西部 1、us-west-1、美國南部(ca-central-1, 中央).

以下是一些 Amazon MSK Replicator 的常見用途。

- 建置多區域串流應用程式：建置高可用性和容錯的串流應用程式，以提高彈性，而無需設定自訂解決方案。
- 更低延遲的資料存取：為不同地理區域的取用者提供較低延遲的資料存取。
- 將資料分發給合作夥伴：將資料從一個 Apache Kafka 叢集複製到多個 Apache Kafka 叢集，以便不同的團隊/合作夥伴擁有自己的資料副本。
- 彙總資料以進行分析：將多個 Apache Kafka 叢集的資料複製到一個叢集，以便輕鬆產生彙總即時資料的洞見資訊。

- 在本機寫入，全域存取您的資料：設定多重作用中複寫，將在一個 AWS 區域中執行的寫入自動傳播到其他區域，以較低的延遲和成本提供資料。

Amazon MSK Replicator 的運作方式

若要開始使用 MSK 複製器，您需要在目標叢集的區域中建立新的複製器。 AWS MSK 複製器會自動將所有資料從叢集中稱為 source 的主要 AWS 區域複製到稱為 target 的目的地區域中的叢集。來源和目標叢集可以位於相同或不同的 AWS 區域中。如果目標叢集並未存在，您將需要建立目標叢集。

建立複製器時，MSK 複製器會在目標叢集的 AWS 區域中部署所有必要的資源，以最佳化資料複寫延遲。複寫延遲會因許多因素而有所不同，包括 MSK 叢集 AWS 區域之間的網路距離、來源和目標叢集的輸送量容量，以及來源和目標叢集上的磁碟分割數目。MSK Replicator 會自動擴充基礎資源，讓您可以隨需複寫資料，而無需監控或擴展容量。

資料複製

根據預設，MSK 複製器會以非同步方式將來源叢集主題磁碟分割中的最新偏移量中的所有資料複製到目標叢集。如果開啟了「偵測並複製新主題」設定，MSK Replicator 會自動偵測新主題或主題磁碟分割，並將其複製到目標叢集。不過，複製器最多可能需要 30 秒的時間才能偵測並建立目標叢集上的新主題或主題磁碟分割。在目標叢集上建立主題之前，對來源主題產生的任何訊息都不會被複寫。或者，如果您想要將主題上的現有訊息複製到目標叢集，則可以在建立期間設定複製器，以從來源叢集主題磁碟分割中的最早偏移量開始複製。

MSK 複製器不會儲存您的資料。資料會從來源叢集中消耗、在記憶體中緩衝，並寫入目標叢集。當資料成功寫入或重試後失敗時，緩衝區會自動清除。MSK 複製器與叢集之間的所有通訊和資料一律會在傳輸過程中加密。所有 MSK 複製器 API 呼叫 (例如 `DescribeClusterV2`、`CreateTopic`、`DescribeTopicDynamicConfiguration` 都會擷取。 AWS CloudTrail您的 MSK 代理程式記錄也會反映相同的內容。

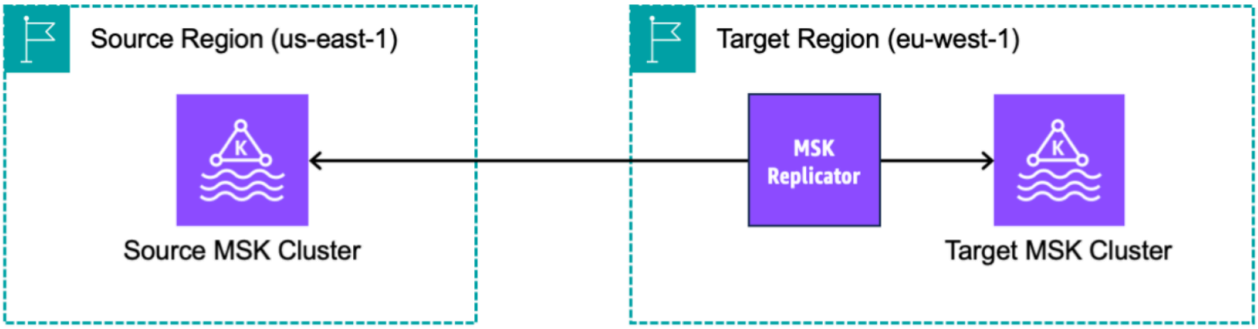
MSK 複製器會在複寫器因子為 3 的目標叢集中建立主題。如有需要，您可以直接在目標叢集上修改複寫因子。

元數據複製

MSK 複製器也支援將中繼資料從來源叢集複製到目標叢集。中繼資料包括主題設定、讀取存取控制清單 (ACL) 和用戶群組偏移。如同資料複寫，中繼資料複寫也會以非同步方式進行 為了獲得更好的效能，MSK 複製器會優先考慮資料複製而不是中繼資料複寫。

作為用戶群組的一部分，MSK Replicator 會針對來源叢集上的取用者進行最佳化，這些用戶從接近串流尖端的位置讀取 (主題分割區結尾)。如果您的用戶群組落後於來源叢集上，與來源相比，您可能會看

到目標上這些用戶群組的延遲較高。這意味著在容錯移轉到目標叢集之後，您的取用者將重新處理更多重複的訊息。為了減少此延遲，來源叢集上的消費者需要 catch 並從串流尖端開始使用 (主題磁碟分割的結尾)。隨著您的消費者 catch 趨勢，MSK 複製器將自動減少延遲。



建立 Amazon MSK Replicator 的要求與考量

請注意這些 MSK 叢集要求適用於執行 Amazon MSK Replicator。

主題

- 建立 MSK Replicator 的必要許可
- 支援的叢集類型和版本
- MSK Serverless 叢集組態
- 叢集組態變更

建立 MSK Replicator 的必要許可

以下是建立 MSK Replicator 所需的 IAM 政策範例。只有在建立 MSK Replicator 時提供了標籤的情況下，才需要動作 `kafka:TagResource`。

```
{
    "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": "*"
        }
    ]
}
```

下方為描述複寫器的 IAM 政策範例。只需要 `kafka:DescribeReplicator` 行動或 `kafka:ListTagsForResource` 行動，而兩者都需要。

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

支援的叢集類型和版本

這些是支援的執行個體類型、Kafka 版本和網路組態的要求。

- MSK Replicator 支援 MSK 佈建叢集和 MSK 無伺服器叢集，兩種可以任合形式組合，作為來源和目標叢集。MSK Replicator 目前不支援其他類型的 Kafka 叢集。
- MSK Serverless 叢集需要 IAM 存取控制，不支援 Apache Kafka ACL 複寫，而且對主題組態複寫提供有限的支援。請參閱MSK Serverless。

- MSK 複製器僅在執行 Apache Kafka 2.7.0 或更高版本的叢集上受支援，無論您的來源和目標叢集位於相同或位於不同區域。 AWS
- MSK Replicator 支援使用 m5.large 或更大執行個體類型的叢集。不支援 t3.small 叢集。
- 如果將 MSK Replicator 與 MSK 佈建叢集搭配使用，則來源和目標叢集中至少要有三個代理程式。您可以在兩個可用區域的所有叢集中複寫資料，但在這些叢集中至少要有四個代理程式。
- 來源和目標 MSK 叢集都必須位於相同的 AWS 帳戶中。不支援不同帳戶中叢集間的複寫。
- 如果來源和目標 MSK 叢集位於不同的區 AWS 域 (跨區域)，MSK 複製器會要求來源叢集為其 IAM 存取控制方法開啟多 VPC 私有連線。來源叢集上的其他身分驗證方法不需要多 VPC。如果您要在相同區域的叢集之間複製資料，則不需要使用多個 VPC。 AWS 請參閱the section called “單一區域多 VPC 私有連線”。

MSK Serverless 叢集組態

- MSK Serverless 支援在主題建立期間複寫 MSK Serverless 目標叢集的下列主題組態：`cleanup.policy`、`compression.type`、`max.message.bytes`、`retention.bytes`、`retenti`
- MSK Serverless 在主題組態同步期間僅支援下列主題組態：`compression.type`、`max.message.bytes`、`retention.bytes`、`retention.ms`。
- 複寫器會在目標 MSK Serverless 叢集上使用 83 個壓縮分區。確定目標 MSK Serverless 叢集具有足夠數量的壓縮分區。請參閱MSK Serverless 配額。

叢集組態變更

- 建議不要在建立 MSK Replicator 之後開啟或關閉分層儲存。如果您的目標叢集沒有啟用分層儲存，則 MSK 不會複製分層儲存組態，無論您的來源叢集是否已啟用分層儲存。如果您在建立複寫器之後，在目標叢集上開啟分層儲存，則需要重新建立複寫器。如果要將資料從未啟用分層儲存的叢集複製到已啟用分層儲存的叢集，則不應該複製主題組態。請參閱啟用和停用現有主題上的分層儲存。
- 在建立 MSK Replicator 之後，請勿變更叢集組態設定。叢集組態設定會在建立 MSK Replicator 期間進行驗證。若要避免 MSK Replicator 發生問題，請勿在建立 MSK Replicator 之後變更下列設定。
  - 將 MSK 叢集變更為 t3 執行個體類型。
  - 變更服務執行角色許可。
  - 停用 MSK 多 VPC 私有連線。
  - 變更連接的叢集資源型政策。
  - 變更叢集安全群組規則。

開始使用 Amazon MSK Replicator

本教學課程說明如何在相同區域或不同 AWS 區 AWS 域中設定來源叢集和目標類別程式。然後您可以使用這些叢集來建立 Amazon MSK Replicator。

步驟 1：準備 Amazon MSK 來源叢集

如果您已經為 MSK Replicator 建立 MSK 來源叢集，請確定其符合本節所描述的要求。否則，請依照下列步驟建立 MSK 佈建或無伺服器來源叢集。

建立跨區域和相同區域 MSK Replicator 來源叢集的程序相似。差異會在下列程序中指明。

1. 在來源區域中開啟 IAM 存取控制的情況下，建立 MSK 佈建或無伺服器叢集。您的來源叢集必須至少要有三個代理程式。
2. 對於跨區域 MSK Replicator，如果來源是佈建叢集，請設定為針對 IAM 存取控制機制開啟多 VPC 私有連線。請注意，開啟多 VPC 後，不支援未驗證的身分驗證類型。您不需要針對其他身分驗證機制 (MTL 或 SASL/SCRAM) 開啟多 VPC 私有連線。您可以針對連線至 MSK 叢集的其他用戶端，同時使用 mTLS 或 SASL/SCRAM 身分驗證機制。您可以在主控台叢集詳細資訊網路設定中或使用 `UpdateConnectivity` API 設定多 VPC 私有連線。請參閱叢集擁有者開啟多 VPC。如果您的來源叢集為 MSK Serverless 叢集，則不需要開啟多 VPC 私有連線。

   對於相同區域 MSK Replicator，MSK 來源叢集不需要多 VPC 私有連線，而且使用未驗證之身分驗證類型的其他用戶端仍可存取叢集。
3. 對於跨區域 MSK Replicator，您必須將資源型許可政策連接至來源叢集。這會允許 MSK 連線到此叢集以複寫資料。您可以使用下面的 CLI 或 AWS 控制台程序執行此操作。亦請參閱 Amazon MSK 資源型政策。您不需要針對相同區域 MSK Replicator 執行此步驟。

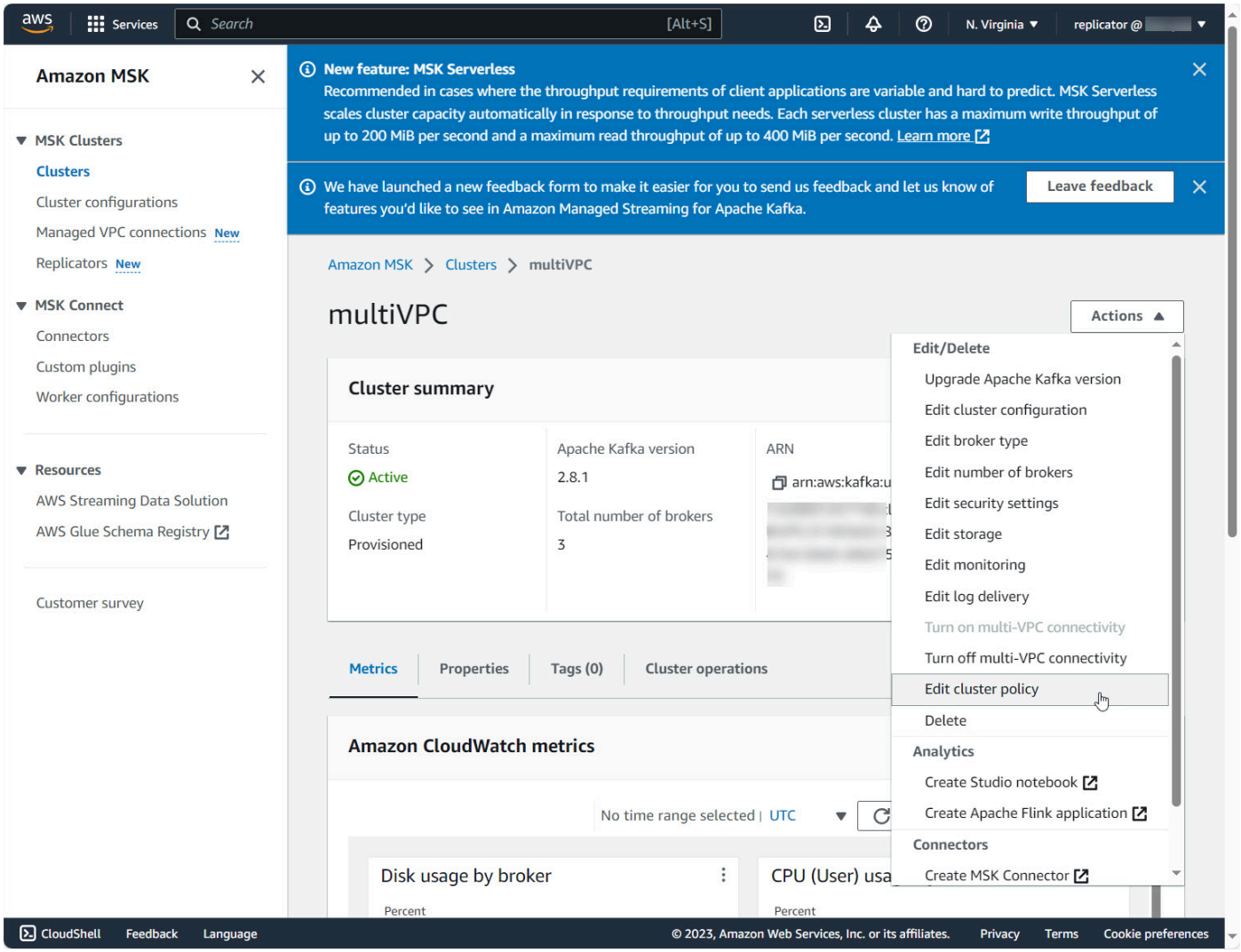
Console: create resource policy

使用下列 JSON 更新來源叢集政策。使用來源叢集的 ARN 取代預留位置。

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

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

在叢集詳細資訊頁面上，使用動作選單下的編輯叢集政策選項。



CLI: create resource policy

備註：如果您使用 AWS 主控台建立來源叢集，並選擇建立新 IAM 角色的選項，請將必要的信任政策 AWS 附加到該角色。如果您想要 MSK 使用現有的 IAM 角色，或您要自行建立角色，請將以下信任政策連接到該角色，以便 MSK Replicator 可以擔任此角色。如需有關如何修改角色之信任關係的資訊，請參閱修改角色。

1. 使用此命令取得目前的 MSK 叢集政策版本。使用實際叢集 ARN 取代預留位置。

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

2. 建立資源型政策，以允許 MSK Replicator 存取您的來源叢集。使用下列語法作為範本，以實際來源叢集 ARN 取代預留位置。

```
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>"
}
]
```

步驟 2：準備 Amazon MSK 目標叢集

建立 MSK 目標叢集 (佈建或無伺服器類型)，並開啟 IAM 存取控制。目標叢集不需要開啟多 VPC 私有連線。目標叢集可以位於與來源叢集相同 AWS 的區域或不同的區域中。來源叢集和目標叢集都必須位於相同的 AWS 帳戶中。您的目標叢集必須至少要有三個代理程式。

步驟 3：建立 Amazon MSK Replicator

建立 Amazon MSK Replicator 之前，請先確定您有 建立 MSK Replicator 的必要許可。

主題

使用 AWS 主控台在目標叢集區域中建立複寫器

1. 在目標 MSK 叢集所在的 AWS 區域中，開啟 Amazon MSK 主控台，網址為 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。
2. 選擇複寫器以顯示帳戶中的複寫器清單。
3. 選擇建立複寫器。
4. 在複寫器詳細資訊窗格中，為新複寫器提供唯一名稱。

選擇來源叢集

來源叢集包含您要複製到目標 MSK 叢集的資料。

1. 在來源叢集窗格中，選擇來源叢集所在的 AWS 區域。

   您可以前往 MSK 叢集並查看叢集詳細資訊 ARN 以查詢叢集的區域。區域名稱內嵌在 ARN 字串中。在下列範例 ARN 中，`ap-southeast-2` 為叢集區域。

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

2. 輸入來源叢集的 ARN，或瀏覽以選擇來源叢集。

3. 選擇來源叢集的子網路。

   主控台會顯示來源叢集區域中可用的子網路，以供您選取。您必須至少選取兩個子網路。對於相同區域 MSK Replicator，您選取設定用來存取來源叢集的子網路，以及用來存取目標叢集的子網路必須位於相同的可用區域之中。

4. 選擇 MSK Replicator 的安全群組，以存取來源叢集。

   - 對於跨區域複寫 (CRR)，您不需要為來源叢集提供安全性群組。
   - 對於相同區域複寫 (SRR)，請前往 Amazon EC2 主控台 https://console.aws.amazon.com/ec2/，確保您將為複製器提供的安全群組具有輸出規則，以允許流量傳輸到來源叢集的安全群組。此外，請確定來源叢集的安全性群組具有輸入規則，允許來自為來源提供的複製器安全性群組的流量。

     若要將輸入規則新增至來源叢集的安全性群組：

     1. 在 AWS 主控台中，選取叢集名稱，移至來源叢集的詳細資料。
     2. 選取屬性索引標籤，然後向下捲動至網路設定窗格，以選取套用的安全群組名稱。
     3. 前往傳入規則，然後選取編輯傳入規則。
     4. 選取新增規則。
     5. 在新規則的 [類型] 欄中，選取 [自訂 TCP]。
     6. 在連接埠範圍欄中，鍵入9098。MSK 複製器使用 IAM 存取控制來連線到使用連接埠 9098 的叢集。
     7. 在來源資料欄中，輸入您在為來源叢集建立複製器期間要提供的安全性群組名稱 (這可能與 MSK 來源叢集的安全性群組相同)，然後選取 [儲存規則]。

     若要將輸出規則新增至為來源提供的複寫器安全性群組：

     1. 在 Amazon EC2 的 AWS 主控台中，前往您將在為來源建立複寫器期間提供的安全群組。
     2. 移至輸出規則，然後選取 [編輯輸出規則]。
     3. 選取新增規則。
     4. 在新規則的 [類型] 欄中，選取 [自訂 TCP]。
     5. 在連接埠範圍欄中，鍵入9098。MSK 複製器使用 IAM 存取控制來連線到使用連接埠 9098 的叢集。
     6. 在 [來源] 資料欄中，輸入 MSK 來源叢集安全性群組的名稱，然後選取 [儲存規則]。

> **Note**
>
> 或者，如果您不想限制使用安全性群組的流量，您可以新增允許所有流量的入站和輸出規則。
> 1. 選取新增規則。
> 2. 在類型欄中，選取所有流量。
> 3. 在來源資料欄中，輸入 `0.0.0.0/0`，然後選取儲存規則。

選擇目標叢集

目標叢集是作為來源資料複製目標的 MSK 佈建或無伺服器叢集。

> **Note**
>
> MSK Replicator 會在目標叢集中建立新主題，並在主題名稱中新增自動產生的字首。例如，MSK Replicator 會將 `topic` 中的資料從來源叢集複寫到目標叢集中名為 `<sourceKafkaClusterAlias>.topic` 的新主題。這是為了區分包含從來源叢集複寫之資料的主題，與目標叢集中的其他主題，並避免在叢集之間循環複寫資料。您可以使用 `DescribeReplicator` API 或 MSK 主控台上的 [複寫器詳細資料] 頁面，在 [sourceKafkaCluster別名] 欄位下找到要新增至目標叢集中主題名稱的前置詞。目標叢集中的前置詞是 < sourceKafkaCluster 別名 >。

1. 在「目標叢集」窗格中，選擇目標叢集所在的 AWS 區域。
2. 輸入目標叢集的 ARN，或瀏覽以選擇目標叢集。
3. 選擇目標叢集的子網路。

   主控台會顯示目標叢集區域中可用的子網路，以供您選取。選取至少兩個子網路。
4. 選擇 MSK Replicator 的安全群組，以存取目標叢集。

   系統將顯示目標叢集區域中可用的安全群組，以供您選取。所選的安全群組會與每個連線相關聯。如需使用安全群組的詳細資訊，請參閱 Amazon VPC 使用者指南中的使用安全群組控制 AWS 資源流量。

   - 對於跨區域複寫 (CRR) 和相同區域複寫 (SRR)，請前往 Amazon EC2 主控台 https://console.aws.amazon.com/ec2/，並確保您將提供給複寫器的安全群組具有輸出規則，以允許流量進入目標叢集的安全群組。此外，請確定目標叢集的安全群組具有傳入規則，可接受來自於為目標提供之複寫器安全群組的流量。

若要將輸入規則新增至目標叢集的安全性群組：

1. 在 AWS 主控台中，選取叢集名稱，移至目標叢集的詳細資料。
2. 選取 [內容] 索引標籤，然後向下捲動至 [網路設定] 窗格，以選取套用的 [安全性] 群組的名稱。
3. 前往傳入規則，然後選取編輯傳入規則。
4. 選取新增規則。
5. 在新規則的 [類型] 欄中，選取 [自訂 TCP]。
6. 在連接埠範圍欄中，鍵入9098。MSK 複製器使用 IAM 存取控制來連線到使用連接埠 9098 的叢集。
7. 在來源資料欄中，輸入您在建立目標叢集的複製器期間要提供的安全性群組名稱 (這可能與 MSK 目標叢集的安全性群組相同)，然後選取 [儲存規則]。

若要將輸出規則新增至為目標提供的複寫器安全性群組：

1. 在 AWS 主控台中，移至您要在為目標建立複寫器期間提供的安全性群組。
2. 選取 [內容] 索引標籤，然後向下捲動至 [網路設定] 窗格，以選取套用的 [安全性] 群組的名稱。
3. 移至輸出規則，然後選取 [編輯輸出規則]。
4. 選取新增規則。
5. 在新規則的 [類型] 欄中，選取 [自訂 TCP]。
6. 在連接埠範圍欄中，鍵入9098。MSK 複製器使用 IAM 存取控制來連線到使用連接埠 9098 的叢集。
7. 在來源資料欄中，輸入 MSK 目標叢集安全性群組的名稱，然後選取儲存規則。

> **Note**
>
> 或者，如果您不想限制使用安全性群組的流量，您可以新增允許所有流量的入站和輸出規則。
> 1. 選取新增規則。
> 2. 在類型欄中，選取所有流量。
> 3. 在來源資料欄中，輸入 `0.0.0.0/0`，然後選取儲存規則。

設定複寫器設定和許可

1. 在複寫器設定窗格中，指定您要使用允許和拒絕清單中的規則運算式複寫的主題。依預設，會複寫所有主題。

   > **Note**
   >
   > MSK 複製器最多只能以排序順序複製 750 個主題。如果您需要複製更多主題，建議您建立個別的複寫器。如果您需要每個複製器超過 750 個主題的支援，請前往主 AWS 控台支援中心並建立支援案例。您可以使用 "TopicCount" 測量結果監督複製的主題數目。請參閱Amazon MSK 配額。

2. 依預設，MSK 複製器會從所選主題中的最新 (最新) 偏移量開始複製。或者，如果您想要複製有關主題的現有資料，可以從所選主題中的最早 (最舊) 偏移量開始複製。建立複製器之後，您就無法變更此設定。此設定對應於`CreateReplicator`要求和`DescribeReplicator`回應 API 中的`startingPosition`欄位。

   > **Note**
   >
   > MSK 複製器就像來源叢集的新取用者一樣。根據您要複製的資料量以及來源叢集中的耗用容量而定，這可能會導致來源叢集上的其他用戶受到限制。如果您將複製器設定為最早的起始位置，MSK 複製器會在開始時讀取大量資料，這可能會消耗來源叢集的所有消耗容量。複製器趕上之後，消耗率應該會下降，以符合來源叢集主題的輸送量。如果您是從最早的位置進行複製，建議您使用 Kafka 配額來管理複製器輸送量，以確保其他消費者不會受到限制。

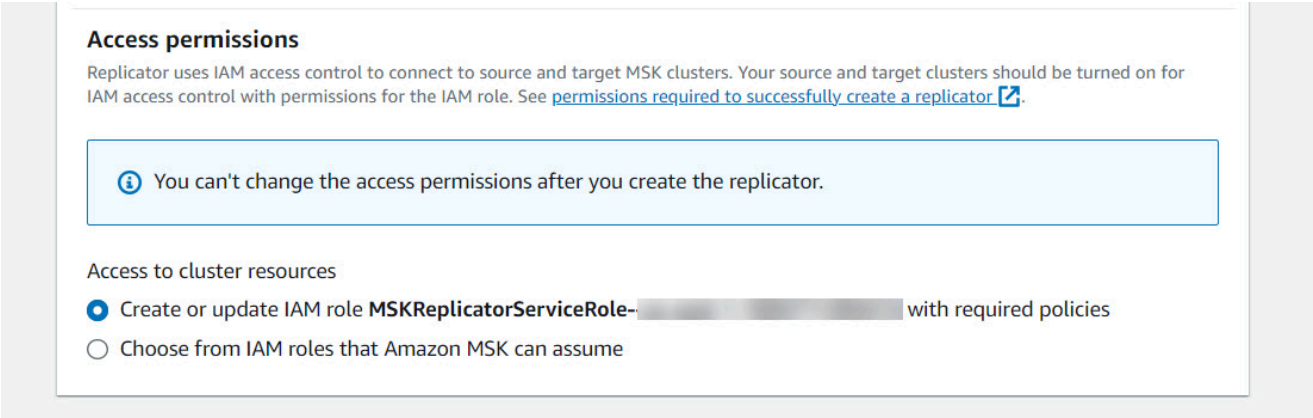
3. 依預設，MSK 複寫器會複製所有中繼資料，包括主題組態、存取控制清單 (ACL) 和取用者群組位移，以實現順暢的容錯移轉。如果您不是建立用於容錯移轉的複寫器，您可以選擇關閉在其他設定區段中的一或多個可用設定。

   > **Note**
   >
   > MSK 複寫器不會複寫寫入 ACL，因為您的生產者不應該直接寫入目標叢集中的複寫主題。容錯移轉後，您的生產者應該寫入目標叢集中的本機主題。如需詳細資訊，請參閱 執行規劃的容錯移轉至次要 AWS 區域。

4. 在取用者群組複寫窗格中，指定您要使用允許和拒絕清單中的規則運算式複寫的取用者群組。依預設，會複寫所有取用者群組。

5. 在壓縮窗格中，您可以選擇壓縮寫入目標叢集的資料。如果要使用壓縮，建議您使用與來源叢集中的資料相同的壓縮方法。
6. 在存取許可窗格中，執行下列其中一項操作：

   a. 選取使用必要政策建立或更新 IAM 角色。MSK 主控台會自動將必要的許可和信任政策連接到讀取和寫入來源和目標 MSK 叢集所需的服務執行角色。



   b. 透過選取 Amazon MSK 可承擔的 IAM 角色中選取 [選擇]，以提供您自己的 IAM 角色。我們建議您將`AWSMSKReplicatorExecutionRole`受管 IAM 政策附加到服務執行角色，而不是撰寫自己的 IAM 政策。

      - 使用以下 JSON 作為信任政策的一部分以及連接到角色的 `AWSMSKReplicatorExecutionRole`，建立複寫器將用於讀取和寫入來源和目標 MSK 叢集的 IAM 角色。在信任政策中，使用您的實際帳戶 ID 取代預留位置 <yourAccountID>。

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

```
    ]
}
```

7. 在複寫器標籤窗格中，您可以選擇將標籤指派給 MSK Replicator 資源。如需詳細資訊，請參閱 標記 Amazon MSK 叢集。對於跨區域 MSK Replicator，在建立複寫器時，會自動將標籤同步至遠端區域。如果您在複寫器建立之後變更標籤，則變更不會自動同步至遠端區域，因此您需要手動同步本機複寫器和遠端複寫器參考資料。
8. 選取建立。

如果您想要限制`kafka-cluster:WriteData`權限，請參閱 Amazon MSK 的 IAM 存取控制如何運作的「建立授權政策」一節。您需要為來源和目標叢集新增`kafka-cluster:WriteDataIdempotently`權限。

MSK Replicator 大約需要 30 分鐘才能成功建立，並轉換為 RUNNING (執行中) 狀態。

如果您建立新的 MSK Replicator 來取代已刪除的複寫器，則新的複寫器會從最新的偏移開始複寫。

如果 MSK Replicator 已轉換為 FAILED (失敗) 狀態，請參閱疑難排解章節的疑難排解 MSK Replicator。

編輯 MSK Replicator 設定

MSK 複製器建立之後，您就無法變更來源叢集、目標叢集或複製器的起始位置。不過，您可以編輯其他複寫器設定，例如要複製的主題和用戶群組。

1. 登入 AWS Management Console，然後開啟 Amazon MSK 主控台，網址為 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。
2. 在左側導覽窗格中，選擇複寫器以顯示帳戶中的複寫器清單，然後選取要編輯的 MSK Replicator。
3. 選擇屬性索引標籤。
4. 在複寫器設定區段中，選擇編輯複寫器。
5. 您可以變更任何設定以編輯 MSK Replicator 設定。
    - 指定您要使用允許和拒絕清單中的規則運算式複寫的主題。依預設，MSK 複寫器會複製所有中繼資料，包括主題組態、存取控制清單 (ACL) 和取用者群組位移，以實現順暢的容錯移轉。如果您不是建立用於容錯移轉的複寫器，您可以選擇關閉在其他設定區段中的一或多個可用設定。

> **Note**
>
> MSK 複寫器不會複寫寫入 ACL，因為您的生產者不應該直接寫入目標叢集中的複寫主題。容錯移轉後，您的生產者應該寫入目標叢集中的本機主題。如需詳細資訊，請參閱執行規劃的容錯移轉至次要 AWS 區域。

- 針對取用者群組複寫，您可以指定要使用允許和拒絕清單中的規則運算式複寫的取用者群組。依預設，會複寫所有取用者群組。如果允許和拒絕清單為空白，則會關閉取用者群組複寫。
- 在目標壓縮類型下，您可以選擇是否要壓縮寫入目標叢集的資料。如果要使用壓縮，建議您使用與來源叢集中的資料相同的壓縮方法。

6. 儲存您的變更。

   MSK Replicator 大約需要 30 分鐘才能成功建立，並轉換為執行中狀態。如果您的 MSK Replicator 已轉換為 FAILED (失敗) 狀態，請參閱疑難排解章節 ???。

刪除 MSK Replicator

如果 MSK Replicator 無法建立 (FAILED 狀態)，那麼您可能需要刪除 MSK Replicator。建立 MSK Replicator 後，無法變更指派給 MSK Replicator 的來源和目標叢集。您可以刪除現有的 MSK Replicator，並建立新的 MSK Replicator。如果您建立新的 MSK Replicator 來取代刪除的複寫器，則新的複寫器會從最新的偏移開始複寫。

1. 在來源叢集所在的 AWS 區域中，登入 AWS Management Console，然後開啟 Amazon MSK 主控台，網址為 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。
2. 在導覽窗格中，選取複寫器。
3. 從 MSK Replicator 清單中，選取您要刪除的複寫器，然後選擇刪除。

監控複寫

您可以在目標叢集區域中使用 https://console.aws.amazon.com/cloudwatch/ 來檢視每個 Amazon MSK Replicator 在主題和彙總層級的 `ReplicationLatency`、`MessageLag`、`ReplicatorThroughput` 指標。度量是ReplicatorName在「AWS/卡夫卡」命名空間下可見。您還可以查看 `ReplicatorFailure`、`AuthError` 和 `ThrottleTime` 指標以檢查問題。

MSK 主控台會顯示每個 MSK 複製器的 CloudWatch 度量子集。從主控台複寫器清單中，選取複寫器的名稱，然後選取監控索引標籤。

MSK Replicator 指標

下列指標描述 MSK Replicator 的效能或連線指標。

AuthError 指標不包括主題級身份驗證錯誤。若要監視 MSK 複製器的主題層級驗證錯誤，請監視複製器的指標以及來源叢集的主題層級 ReplicationLatency 指標。 MessagesInPerSec如果某個主題被 ReplicationLatency 丟棄到 0，但該主題仍然有數據正在產生，則表明複製器與該主題存在身份驗證問題。檢查複寫器的服務執行 IAM 角色是否具有足夠的許可來存取該主題。

指標類型	指標	描述	維度	單位	原始指標精細程度	原始指標彙總統計資料	
效能	ReplicationLatency	將記錄從來源複寫至目標叢集所需的時間；從在來源產生記錄到複寫至目標之間的持續時間。如果 ReplicationLatency 增加，請檢查叢集是否有足夠的分割區來支援複寫。當分區計數過低而無法達到高輸送量時，可能會發生較長的複寫延遲。	ReplicatorName	毫秒	分區	最大	
			ReplicatorName, 主題	毫秒	分區	最大	
效能	MessageLag	監視 MSK 複製器和來源叢集之間的同步。MessageLag 指出產生給來源叢集的訊息與複製器使用的訊息	ReplicatorName	計數	分區	總和	
			ReplicatorName, 主題	計數	分區	總和	

指標類型	指標	描述	維度	單位	原始指標精細程度	原始指標彙總統計資料	
		之間的延遲。它不是源和目標集群之間的滯後。即使來源叢集不可用/中斷，複製器也會完成將已耗用的訊息寫入目標叢集。中斷後，會 MessageLag 顯示一個增加，表示複寫器位於來源叢集後方的訊息數目，而且可以監視這個數目，直到訊息數目為 0，表示複寫器已趕上來源叢集。					

指標類型	指標	描述	維度	單位	原始指標精細程度	原始指標彙總統計資料	
效能	ReplicatorThroughput	每秒平均複寫的位元組數量。如果主題中 ReplicatorThroughput 斷，請檢查 KafkaClusterPingSuccessCount 和 AuthError 測量結果以確保複製器可以與叢集通訊，然後檢查叢集指標以確保叢集未關閉。	ReplicatorName	BytesPerSecond	分區	總和	
			ReplicatorName, 主題	BytesPerSecond	分區	總和	
偵錯	AuthError	每秒身分驗證失敗的連線數量。如果此指標超過 0，您可以檢查複寫器的服務執行角色政策是否有效，並確定未針對叢集設定拒絕許可。根據 clusterAlias 維度，您可以識別來源或目標叢集是否遇到身分驗證錯誤。	ReplicatorName, ClusterAlias	計數	工作程序	總和	

指標類型	指標	描述	維度	單位	原始指標精細程度	原始指標彙總統計資料	
偵錯	ThrottleTime	叢集代理程式限流請求的平均時間 (毫秒)。設定限流以避免 MSK Replicator 使叢集不堪負荷。如果此指標為 0，而 replicationLatency 不高，且 replicatorThroughput 符合預期，則限流會如預期般運作。如果此指標大於 0，您可以相應地調整限流。	ReplicatorName, ClusterAlias	毫秒	工作程序	最大	
偵錯	ReplicatorFailure	複寫器發生的失敗次數。	ReplicatorName	計數		總和	

指標類型	指標	描述	維度	單位	原始指標精細程度	原始指標彙總統計資料	
偵錯	KafkaClusterPingSuccessCount	指出複寫器與 kafka 叢集之連線的運作狀態。如果此值為 1，表示連線的運作狀態良好。如果此值為 0 或沒有資料點，表示連線的運作狀態不佳。如果此值為 0，請檢查 Kafka 叢集的網路或 IAM 許可設定。您可以根據 ClusterAlias 維度來識別此測量結果是用於來源叢集還是目標叢集。	ReplicatorName, ClusterAlias	計數		總和	

使用複寫提高跨區域之 Kafka 串流應用程式的彈性

您可以使用 MSK 複製器來設定主動-主動或主動-被動叢集拓撲，以提高跨區域之 Apache Kafka 應用程式的彈性。 AWS 在主動-主動式設定中，兩個 MSK 叢集都主動提供讀取和寫入。在主動-被動式設定中，一次只有一個 MSK 叢集主動提供串流資料，而另一個叢集處於待命狀態。

建置多區域 Apache Kafka 應用程式的考量事項

您的取用者必須能夠重新處理重複的訊息，且不影響下游。MSK 複製器會複製資料 at-least-once ，這可能會導致待命叢集中出現重複的資料。當您切換到次要 AWS 區域時，您的消費者可能會多次處理相同的資料。MSK Replicator 會將資料複製作業優先於取用者偏移，以取得更好的效能。容錯移轉之後，取用者可能會從較早的偏移開始讀取，進而導致重複處理。

生產者和取用者還必須容忍失去少量的資料。由於 MSK 複製器會以非同步方式複寫資料，因此當主要 AWS 區域開始發生故障時，無法保證所有資料都會複寫到次要區域。您可以使用複寫延遲，以判斷未複製到次要區域的資料大小上限。

主動-主動式與主動-被動式叢集拓撲比較

主動-主動式叢集拓撲提供接近零的復原時間，而且可讓串流應用程式在多個 AWS 區域同時運作。當一個區域中的叢集受損時，連線到另一個區域中叢集的應用程式會繼續處理資料。

主動-被動式設定適用於一次只能在一個 AWS 區域中執行的應用程式，或當您需要對資料處理順序有更多的控制時。主動-被動式設定比主動-主動式設定需要更多的復原時間，因為您必須在次要區域啟動整個主動-被動式設定，包括您的生產者和取用者，才能在容錯移轉後恢復串流資料。

建立主動-被動式 Kafka 叢集設定和複寫主題命名

對於主動-被動設定，建議您在兩個不同的區域中操作類似的生產者、MSK 叢集和用戶 (具有相同用戶群組名稱) 的設定。 AWS 重要的是，兩個 MSK 叢集具有相同的讀取和寫入容量，以確保可靠的資料複寫。您需要建立 MSK Replicator，以持續從主要叢集將資料複製到待命叢集。您也需要設定生產者將資料寫入相同 AWS 區域中叢集上的主題。

為了確保您的取用者能夠可靠地在待命叢集上重新啟動處理，您需要將取用者設定為使用萬用字元運算子 ".*" 從主題中讀取資料。例如，MSK 複製器會將「主題 1」從主要叢集複寫到待命叢集中的新主題，稱為「< 別名 >.Topic1」。sourceKafkaCluster例如，您可以將生產者設定為寫入 "topic1"，並將取用者設定為在兩個地區中使用 ".*topic1" 進行取用。此範例也會包含諸如 footopic1 之類的主題，因此請根據您的需要調整萬用字元運算子。

何時容錯移轉至次要 AWS 區域

我們建議您使用 CloudWatch. AWS 在主要 AWS 區域中發生服務事件期間，複寫延遲可能會突然增加。如果延遲不斷增加，請使用 AWS Service Health Dashboard 來檢查主要 AWS 區域中的服務事件。如果發生事件，您可以容錯移轉至次要 AWS 區域。

執行規劃的容錯移轉至次要 AWS 區域

您可以執行規劃的容錯移轉，以針對具有來源 MSK 叢集的主要 AWS 區域中的意外事件來測試應用程式的復原能力。規劃的容錯移轉不會導致資料遺失。

1. 關閉連線至來源叢集的所有生產者和取用者。

2. 建立新的 MSK Replicator，從次要區域中的 MSK 叢集將資料複寫至主要區域中的 MSK 叢集。如果需要將要寫入次要區域的資料複寫回主要區域，以便在意外事件結束後容錯恢復至主要區域，就需要此操作。
3. 在次要 AWS 區域中的目標叢集上啟動生產者。
4. 根據應用程式的訊息順序要求而定，依照下列其中一個索引標籤中的步驟進行操作。

No message ordering

如果您的應用程式不需要訊息排序，請啟動次要 AWS 區域中使用萬用字元運算子 (例如，`topic`) 同時讀取本機 (例如`<sourceKafkaClusterAlias>.topic`) 和複製主題 (例如) 的取用者 (例如，.* 主題）。

Message ordering

如果您的應用程式需要訊息排序，請僅針對目標叢集 (例如 `<sourceKafkaClusterAlias>.topic`) 上的複寫主題，而非本機主題 (例如，`topic`) 啟動取用者。

1. 等待目標 MSK 叢集上複寫主題的所有取用者完成處理所有資料，取用者延遲為 0，且處理的記錄數量也為 0。然後，停止目標叢集上複製主題的取用者。此時，已取用從來源 MSK 叢集複寫到目標 MSK 叢集的所有記錄。
2. 啟動目標 MSK 叢集上本機主題 (例如 `topic`) 的取用者。

執行意外的容錯移轉至次要區域 AWS

當具有來源 MSK 叢集的主要 AWS 區域中發生服務事件，且您想要暫時將流量重新導向至具有目標 MSK 叢集的次要 AWS 區域時，您可以執行意外的容錯移轉。非計劃容錯移轉可能導致部分資料遺失。

1. 嘗試關閉連線至主要區域中來源 MSK 叢集的所有生產者和取用者。這可能會失敗。
2. 啟動連線至次要區域中目標 MSK 叢集的生產者。
3. 根據應用程式的訊息順序要求而定，依照下列其中一個索引標籤中的步驟進行操作。

   No message ordering

   如果您的應用程式不需要訊息排序，請啟動目標 AWS 區域中使用萬用字元運算子 (例如，`topic`) 同時讀取本機 (例如`<sourceKafkaClusterAlias>.topic`) 和複寫主題 (例如) 的取用者 (例如，`.*topic`)。

Message ordering

1. 請僅針對目標叢集 (例如 <sourceKafkaClusterAlias>.topic) 上的複寫主題，而非本機主題 (例如，topic) 啟動取用者。
2. 等待目標 MSK 叢集上複寫主題的所有取用者完成處理所有資料，偏移延遲為 0，且處理的記錄數量也為 0。然後，停止目標叢集上複製主題的取用者。此時，已取用從來源 MSK 叢集複寫到目標 MSK 叢集的所有記錄。
3. 啟動目標 MSK 叢集上本機主題 (例如 topic) 的取用者。

4. 服務事件在主要區域中結束後，請建立新的 MSK 複製器，將次要區域中的 MSK 叢集中的資料複製到主要區域中的 MSK 叢集，並將複製器起始位置設為最早的主要區域中的 MSK 叢集。如果需要將要寫入次要區域的資料複寫回主要區域，以便在服務事件結束後容錯恢復至主要區域，就需要此操作。如果您未將複製器的起始位置設定為最早，則在主要區域發生服務事件期間，您在次要區域中產生給叢集的任何資料都不會複製回主要區域中的叢集。

執行容錯回復至主要區域 AWS

您可以在該 AWS 區域的服務事件結束後容錯回復至主要區域。在容錯恢復期間，將資料複寫回主要區域時，MSK Replicator 會自動略過以來源叢集別名作為字首的主題。

如果您遵循了意外的容錯移轉步驟，則應該已經建立容錯回復複製器，作為從主要區域容錯移轉到次要區域最後一個步驟的一部分。

如果您沒有遵循意外的容錯移轉步驟，服務事件在主要區域結束後，建立新的 MSK 複製器，將資料從次要區域中的 MSK 叢集複製到主要區域中的 MSK 叢集 (複製器起始位置設為最早)。如果需要將要寫入次要區域的資料複寫回主要區域，以便在服務事件結束後容錯恢復至主要區域，就需要此操作。如果您沒有將複製器的起始位置從預設值最新變更為最早，則在主要區域發生服務事件期間，您在次要區域中產生至叢集的任何資料都不會複製回主要區域中的叢集。

只有在從次要區域中的叢集複寫到主要區域中的叢集，且中的 MessageLag 度量接近 0 之後，您才應啟動容錯回復步驟。 CloudWatch 計劃的容錯恢復不會導致任何資料遺失。

1. 關閉連線至次要區域中 MSK 叢集的所有生產者和取用者。
2. 對於主動-被動式拓撲，刪除從次要區域中叢集將資料複寫到主要區域的複寫器。您不需要刪除主動-主動式拓撲的複寫器。
3. 啟動連線至主要區域中 MSK 叢集的生產者。
4. 根據應用程式的訊息順序要求而定，依照下列其中一個索引標籤中的步驟進行操作。

No message ordering

如果您的應用程式不需要訊息排序，請啟動使用萬用字元運算子 (例如) 同時讀取本機 (例如topic) 和複寫主題 (例如<sourceKafkaClusterAlias>.topic) 的主要 AWS 區域中的取用者 (例如，.*topic)。本機主題 (例如：topic) 上的取用者將從容錯移轉之前取用者取用的最後一個偏移恢復。如果在容錯移轉之前有任何未處理的資料，則將會立即處理該資料。如果是計劃的容錯移轉，應該沒有此類記錄。

Message ordering

1. 請僅針對主要區域 (例如 <sourceKafkaClusterAlias>.topic) 上的複寫主題，而非本機主題 (例如，topic) 啟動取用者。
2. 等待主要區域中叢集上複寫主題的所有取用者完成處理所有資料，偏移延遲為 0，且處理的記錄數量也為 0。然後，停止在主要區域中叢集上複製主題的取用者。此時，容錯移轉後在次要區域中產生的所有記錄皆已在主要區域中取用。
3. 啟動主要區域中叢集上本機主題 (例如 topic) 的取用者。

5. 使用和延遲度量，確認從主要區域到叢集中的現有複製器處於 RUNNING 狀態，ReplicatorThroughput並如預期般運作。

使用 MSK Replicator 建立主動-主動式設定

依照下列步驟，在來源 MSK 叢集 A 與目標 MSK 叢集 B 之間設定主動-主動式拓樸。

1. 建立 MSK Replicator，將 MSK 叢集 A 作為來源，MSK 叢集 B 作為目標。
2. 成功建立上述 MSK Replicator 之後，請建立以叢集 B 作為來源，建立叢集 A 作為目標的複寫器。
3. 建立兩組生產者，兩組生產者可同時將資料寫入在與生產者相同區域中叢集的本機主題 (例如 "topic")。
4. 建立兩組取用者，每個使用者都使用萬用字元訂閱的讀取資料 (例如」. * 主題」) 來自與消費者位於相同 AWS 區域的 MSK 叢集。這樣，您的取用者將自動從本機主題 (例如，topic) 中讀取在區域中本機產生的資料，以及從其他區域帶有字首 <sourceKafkaClusterAlias>.topic 的主題中複寫的資料。這兩組取用者應具有不同的取用者群組 ID，以便在 MSK Replicator 將取用者群組複製到另一個叢集時，不會覆寫取用者群組偏移。

疑難排解 MSK Replicator

主題

下列資訊可協助您針對 MSK Replicator 可能發生的問題進行疑難排解。您也可以將您的問題張貼到 AWS re:Post。

MSK Replicator 狀態從 CREATING (建立中) 變為 FAILED (失敗)

以下是 MSK Replicator 建立失敗的一些常見原因。

1. 驗證您在目標叢集區段中為建立複寫器提供的安全群組是否具有傳出規則，以允許流量前往目標叢集的安全群組。此外，請驗證目標叢集的安全群組是否具有傳入規則，可接受來自於目標叢集區段中為建立複寫器提供之安全群組的流量。請參閱選擇目標叢集。
2. 如果您要建立跨區域複寫的複寫器，請驗證來源叢集是否已針對 IAM 存取控制身分驗證方法開啟多 VPC 連線。請參閱Amazon MSK 的單一區域多 VPC 私有連線。同時驗證是否已在來源叢集上設定叢集政策，以便 MSK Replicator 可以連線到來源叢集。請參閱步驟 1：準備 Amazon MSK 來源叢集。
3. 驗證您在建立 MSK Replicator 期間提供的 IAM 角色是否具有讀取和寫入來源和目標叢集所需的許可。此外，請驗證 IAM 角色是否具有寫入主題的許可。請參閱設定複寫器設定和許可
4. 驗證您的網路 ACL 是否未封鎖 MSK Replicator 與來源和目標叢集之間的連線。
5. MSK Replicator 嘗試連線至來源或目標叢集時，可能無法完全使用來源或目標叢集。這可能是因為負載過多、磁碟使用率或 CPU 使用率過高，造成複寫器無法連線至代理程式。修復代理程式的問題，然後重試建立複寫器。

執行上述驗證後，請再次建立 MSK Replicator。

MSK Replicator 顯示停滯在 CREATING 狀態

有時建立 MSK Replicator 最多需要 30 分鐘。等候 30 分鐘，然後再次檢查複寫器的狀態。

MSK Replicator 未複製資料或僅複製部分資料

依照下列步驟，對資料複寫問題進行疑難排解。

1. 使用中的 MSK 複製器提供的 AuthError 指標，確認您的複製器未發生任何驗證錯誤。 CloudWatch 如果此指標超過 0，請檢查複寫器的 IAM 角色政策是否有效，並且未針對叢集設定拒絕許可。根據 clusterAlias 維度，您可以識別來源或目標叢集是否遇到身分驗證錯誤。
2. 驗證您的來源和目標叢集沒有遇到任何問題。複寫器可能無法連線到來源或目標叢集。這可能是由於連線太多，磁盤容量全滿或高 CPU 使用率。
3. 使用中的 KafkaClusterPingSuccessCount 指標，確認您的來源和目標叢集可從 MSK 複製器存取。CloudWatch根據 clusterAlias 維度，您可以識別來源或目標叢集是否遇到身分驗證錯誤。如果此指標為 0 或沒有資料點，則表示連線的運作狀態不佳。您應該檢查 MSK Replicator 用來連線到叢集的網路和 IAM 角色許可。
4. 使用中的指標，確認您的複製器未因缺少主題層級權限而失敗。 ReplicatorFailure CloudWatch如果此指標高於 0，請檢查您為主題層級許可提供的 IAM 角色。
5. 驗證您在建立複寫器時，在允許清單中提供的規則運算式是否與您要複寫的主題名稱相符。此外，請驗證主題並未因為拒絕清單中的規則運算式而排除在複寫之外。
6. 請注意，複製器最多可能需要 30 秒的時間才能偵測並建立目標叢集上的新主題或主題磁碟分割。如果複製器的起始位置是最新的 (預設值)，則不會複寫在目標叢集上建立主題之前對來源主題產生的任何訊息。或者，如果您想要複寫目標叢集上主題上的現有訊息，則可以從來源叢集主題磁碟分割中的最早偏移量開始複寫。請參閱設定複寫器設定和許可。

目標叢集中的訊息偏移量與來源叢集不同

作為複寫資料的一部分，MSK 複製器會使用來源叢集中的訊息，並將其產生到目標叢集。這可能會導致訊息在來源和目標叢集上具有不同的偏移量。不過，如果您已在複製器建立期間開啟用戶群組偏移同步，MSK Replicator 會在複製中繼資料時自動轉譯偏移，以便在容錯移轉至目標叢集之後，您的取用者可以從來源叢集中停止的地方繼續處理。

MSK 複製器未同步用戶群組偏移，或目標叢集上不存在用戶群組

請依照下列步驟疑難排解中繼資料複寫問題

1. 確認資料複製是否如預期般運作。如果沒有，請參閱 MSK Replicator 未複製資料或僅複製部分資料。

2. 確認您在建立複製器時，在允許清單中提供的規則運算式符合您要複寫之用戶群組的名稱。此外，請確認用戶群組並未因為拒絕清單中的規則運算式而排除在複寫之外。
3. 確認 MSK 複製器已在目標叢集上建立主題。複製器最多可能需要 30 秒的時間才能偵測並建立目標叢集上的新主題或主題磁碟分割。如果複製器的起始位置是最新的 (預設值)，則不會複寫在目標叢集上建立主題之前對來源主題產生的任何訊息。如果來源叢集上的用戶群組只耗用尚未由「MSK 複製器」複寫的訊息，則不會將用戶群組複製到目標叢集。在目標叢集上成功建立主題之後，MSK 複製器會開始將來源叢集上新寫入的訊息複寫到目標。一旦用戶群組開始從來源讀取這些訊息，MSK 複製器就會自動將用戶群組複寫到目標叢集。或者，如果您想要複寫目標叢集上主題上的現有訊息，則可以從來源叢集主題磁碟分割中的最早偏移量開始複寫。請參閱設定複寫器設定和許可。

> **Note**
>
> MSK Replicator 會針對來源叢集上的取用者最佳化用戶群組偏移同步，這些用戶群組會從接近主題磁碟分割結尾的位置讀取。如果您的用戶群組落後於來源叢集上，與來源相比，您可能會看到目標上這些用戶群組的延遲較高。這意味著在容錯移轉到目標叢集之後，您的取用者將重新處理更多重複的訊息。為了減少此延遲，來源叢集上的消費者需要 catch 並從串流尖端開始使用 (主題磁碟分割的結尾)。隨著您的消費者 catch 趨勢，MSK 複製器將自動減少延遲。

複寫延遲很高或持續增加

以下是造成高複寫延遲的一些常見原因。

1. 驗證在來源和目標 MSK 叢集上的分區數量是否正確。分區過少或過多可能會影響效能。如需有關選擇分區數量的指引，請參閱 使用 MSK Replicator 的最佳實務。下表顯示 MSK Replicator 取得所需輸送量建議的分區數量下限。

   輸送量和建議的分區數量下限

輸送量 (MB/s)	所需的分區數量下限
50	167
100	334
250	833
500	1666

輸送量 (MB/s)	所需的分區數量下限
1000	3333

2. 驗證在來源和目標 MSK 叢集中具有足夠的讀取和寫入容量，以支援複寫流量。MSK Replicator 可作為來源叢集 (輸出) 的取用者，以及作為目標叢集 (輸入) 的生產者。因此，除了叢集上的其他流量以外，您還應佈建叢集容量以支援複寫流量。如需有關調整 MSK 叢集大小的指引，請參閱 ???。
3. 不同來源和目的地 AWS 區域配對中 MSK 叢集的複寫延遲可能會有所不同，具體取決於叢集彼此之間的距離。例如，在歐洲 (愛爾蘭) 與亞太區域 (雪梨) 區域中叢集之間的複寫，比在歐洲 (愛爾蘭) 與歐洲 (倫敦) 區域中叢集之間的複寫延遲更低。
4. 驗證您的複寫器沒有因為在來源或目標叢集上設定過度積極的配額而受到限流。您可以使用 MSK 複製器提供的 ThrottleTime測量結果，以查看 CloudWatch 來源/目標叢集上的中介程式限制要求的平均時間 (毫秒)。如果此指標高於 0，則應調整 Kafka 配額以減少限流，以便複寫器可以追上。如需有關管理複寫器 Kafka 配額的資訊，請參閱 使用 Kafka 配額管理 MSK Replicator 輸送量。
5. ReplicationLatency 當一個 AWS 區域變得降級時， MessageLag 可能會增加。使用 AWS 服務運作狀態儀表板 來檢查主要 MSK 叢集所在區域中是否有 MSK 服務事件。如果發生服務事件，您可以暫時將應用程式讀取和寫入重新導向至其他區域。

使用 MSK Replicator 的最佳實務

本節介紹了使用 MSK 複寫器的常見最佳實務與實施策略。

主題

- 使用 Kafka 配額管理 MSK Replicator 輸送量
- 設定叢集保留期間

使用 Kafka 配額管理 MSK Replicator 輸送量

由於 MSK Replicator 可作為來源叢集的取用者，因此複寫可能會導致來源叢集上的其他取用者受到限流。限流量取決於您在來源叢集上擁有的讀取容量，以及要複寫的資料輸送量。我們建議您為來源和目標叢集佈建相同的容量，並在計算您需要的容量時考慮複寫輸送量。

您也可以在來源和目標叢集上的設定複寫器的 Kafka 配額，以控制 MSK Replicator 可以使用的容量。建議使用網路頻寬配額。網路頻寬配額會針對一個或多個共用配額的用戶端，定義的位元組率閾值 (定義為每秒位元組數)。此配額是根據每個代理程式而定義。

請依照下列步驟套用配額。

1. 擷取來源叢集的引導伺服器字串。請參閱取得 Amazon MSK 叢集的引導代理程式。
2. 擷取 MSK Replicator 使用的服務執行角色 (SER)。這是您用於 `CreateReplicator` 請求的 SER。您也可以從現有複寫器的 DescribeReplicator 回應中提取 SER。
3. 使用 Kafka CLI 工具，針對來源叢集執行下列命令。

```
./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. 執行上述命令後，驗證 `ReplicatorThroughput` 指標是否未超過您設定的配額。

請注意，如果您在多個 MSK Replicator 之間重複使用同一個服務執行角色，則它們皆會受到此配額的限制。如果您想要維護每個複寫器的個別配額，請使用個別的服務執行角色。

如需有關將 MSK IAM 身分驗證與配額搭配使用的詳細資訊，請參閱 Multi-tenancy Apache Kafka clusters in Amazon MSK with IAM access control and Kafka Quotas – Part 1。

> **Warning**
>
> 設定極低的 consumer_byte_rate 可能會導致 MSK Replicator 以非預期的方式運作。

設定叢集保留期間

您可以為 MSK 佈建或無伺服器叢集設定日誌保留期間。建議保留期間為 7 天。請參閱 叢集組態變更 或 MSK Serverless 叢集組態。

叢集狀態

下表顯示叢集的可能狀態，並說明其意義。其也會說明當叢集為上述其中一種狀態時，您可以執行和無法執行的動作。若要了解叢集的狀態，您可以前往 AWS Management Console。您也可以使用 describe-cluster-v2 命令或 DescribeClusterV 2 作業來描述叢集。叢集的描述包括其狀態。

叢集狀態	意義和可能行動
ACTIVE	您可以產生和取用資料。您也可以在叢集上執行 Amazon MSK API 和 AWS CLI 操作。
CREATING	Amazon MSK 正在設定叢集。您必須等待叢集到達 ACTIVE 狀態，才能使用叢集產生或使用資料，或對其執行 Amazon MSK API 或 AWS CLI 操作。
DELETING	正在刪除叢集。您無法使用它來產生或取用資料。您也無法在其上執行 Amazon MSK API 或 AWS CLI 操作。
失敗	叢集建立或刪除程序失敗。您無法使用叢集來產生或取用資料。您可以刪除叢集，但無法對叢集執行 Amazon MSK API 或 AWS CLI 更新作業。
HEALING	Amazon MSK 正在執行內部操作，例如取代運作狀態不良的代理程式。例如，代理程式可能沒有回應。您仍可使用叢集來產生和使用資料。不過，您無法在叢集上執行 Amazon MSK API 或 AWS CLI 更新作業，直到它返回到作用中狀態為止。
MAINTENANCE	Amazon MSK 正在叢集上執行例行性維護作業。此類維護作業包括安全性修補。您仍可使用叢集來產生和使用資料。不過，您無法在叢集上執行 Amazon MSK API 或 AWS CLI 更新作業，直到它返回到作用中狀態為止。

叢集狀態	意義和可能行動
REBOOTING_BROKER	Amazon MSK 正在重新啟動代理程式。您仍可使用叢集來產生和使用資料。不過，您無法在叢集上執行 Amazon MSK API 或 AWS CLI 更新作業，直到它返回到作用中狀態為止。
UPDATING	使用者啟動的 Amazon MSK API 或 AWS CLI 作業正在更新叢集。您仍可使用叢集來產生和使用資料。不過，您無法在叢集上執行任何其他 Amazon MSK API 或 AWS CLI 更新作業，直到它返回到作用中狀態為止。

Amazon Managed Streaming for Apache Kafka 的安全性

雲安全 AWS 是最高的優先級。身為 AWS 客戶，您可以從資料中心和網路架構中獲益，該架構專為滿足對安全性最敏感的組織的需求而打造。

安全是 AWS 與您之間共同的責任。共同責任模型將其描述為雲端的安全性和雲端中的安全性：

- 雲端的安全性 — AWS 負責保護在 AWS 雲端中執行 AWS 服務的基礎架構。 AWS 還為您提供可以安全使用的服務。若要了解適用於 Amazon Managed Streaming for Apache Kafka 的合規計畫，請參閱 Amazon Web Services 的合規計劃服務範圍。
- 雲端安全性 — 您的責任取決於您使用的 AWS 服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件有助於您了解如何在使用 Amazon MSK 時套用共同責任模型。下列主題說明如何設定 Amazon MSK 來符合您的安全與合規目標。您也將了解如何使用其他 Amazon Web Services，幫助您監控並保護 Amazon MSK 資源。

主題

- Amazon Managed Streaming for Apache Kafka 的資料保護
- Amazon MSK API 的身分驗證和授權
- Apache Kafka API 的身分驗證和授權
- 變更 Amazon MSK 叢集的安全群組
- 控制對阿帕奇的存取 ZooKeeper
- 日誌
- Amazon Managed Streaming for Apache Kafka 的合規驗證
- Amazon Managed Streaming for Apache Kafka 的復原能力
- Amazon Managed Streaming for Apache Kafka 的基礎設施安全性

Amazon Managed Streaming for Apache Kafka 的資料保護

AWS 共同責任模型適用於 Apache Kafka 的 Amazon 受管串流中的資料保護。如此模型所述， AWS 負責保護執行所有 AWS 雲端. 您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。如需

有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 AWS 共同的責任模型和 GDPR 部落格文章。

基於資料保護目的，我們建議您使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 保護 AWS 帳戶 登入資料並設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源進行通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用設定 API 和使用者活動記錄 AWS CloudTrail。
- 使用 AWS 加密解決方案以及其中的所有默認安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在透過命令列介面或 API 存取時需要經 AWS 過 FIPS 140-2 驗證的加密模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱聯邦資訊處理標準 (FIPS) 140-2 概觀。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用主控台、API 或 AWS 開發套件 AWS 服務 使用 Amazon MSK 或其他工作時。 AWS CLI您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

主題

- Amazon MSK 加密
- 如何開始使用加密？

Amazon MSK 加密

Amazon MSK 提供資料加密選項，您可以使用這些選項來符合嚴格的資料管理需求。Amazon MSK 用於加密的憑證必須每 13 個月更新一次。Amazon MSK 會自動更新所有叢集的這些憑證。啟動憑證更新作業時，它會將叢集的狀態設定為 `MAINTENANCE`。更新完成時，它會將設定調回 `ACTIVE`。當叢集處於 `MAINTENANCE` 狀態時，您可以繼續產生和使用資料，但無法對叢集執行任何更新作業。

靜態加密

Amazon MSK 與 AWS Key Management Service (KMS) 整合，以提供透明的伺服器端加密。Amazon MSK 一律會加密靜態資料。建立 MSK 叢集代理程式時，可以指定想要 Amazon MSK 用來加密靜態資料的 AWS KMS key 。如不指定 KMS 金鑰，Amazon MSK 就會為您建立 AWS 受管金鑰，並代表您

使用它。如需 KMS 金鑰的詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的 AWS KMS keys。

傳輸中加密

Amazon MSK 使用 TLS 1.2。根據預設，會加密您 MSK 叢集代理程式之間傳輸中的資料。您可以在建立叢集時覆寫此預設值。

針對用戶端與代理程式之間的通訊，您必須指定下列三種設定之一：

- 只允許 TLS 加密的資料。這是預設設定。
- 允許純文字，以及 TLS 加密的資料。
- 只允許純文字資料。

Amazon MSK 代理程式使用公有 AWS Certificate Manager 憑證。因此，任何信任 Amazon 信任服務的信任存放區，也會信任 Amazon MSK 代理程式憑證。

我們強烈建議啟用傳輸中加密，但這可能會給 CPU 增加額外的負荷和幾毫秒的延遲。不過，大多數使用案例對這些差異並不敏感，且影響程度取決於叢集、用戶端和使用設定檔的組態。

如何開始使用加密？

建立 MSK 叢集時，可以指定 JSON 格式的加密設定。以下是範例。

```
{
   "EncryptionAtRest": {
       "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/abcdabcd-1234-
abcd-1234-abcd123e8e8e"
    },
   "EncryptionInTransit": {
        "InCluster": true,
        "ClientBroker": "TLS"
    }
}
```

針對 `DataVolumeKMSKeyId`，您可以指定客戶自管金鑰，或為您的帳戶中的 MSK 選擇 AWS 受管金鑰 (`alias/aws/kafka`)。如果您未指定`EncryptionAtRest`，Amazon MSK 仍會加密您在. AWS 受管金鑰若要判斷叢集正在使用哪個金鑰，請傳送 `GET` 請求或調用 `DescribeCluster` API 操作。

針對 `EncryptionInTransit`，`InCluster` 的預設值為 true，但如果您不想要 Amazon MSK 加密在代理程式之間傳遞的資料，則可將其設定為 false。

若要指定用戶端與代理程式之間傳輸資料的加密模式，請將 ClientBroker 設定為下列三個值之一：TLS、TLS_PLAINTEXT、或 PLAINTEXT。

若要在建立叢集時指定加密設定

1. 將前一個範例的內容儲存在檔案中，並為檔案命名為任何您想要的名稱。例如，稱之為 encryption-settings.json。
2. 執行 create-cluster 命令並使用 encryption-info 選項來指定您儲存 JSON 組態的文件。以下是範例。使用 Apache Kafka 用戶端版本取代 *{YOUR MSK VERSION}*。如需有關如何尋找 MSK 叢集版本的資訊，請參閱To find the version of your MSK cluster。請注意，使用與 MSK 叢集版本不同的 Apache Kafka 用戶端版本，可能導致 Apache Kafka 資料損毀、遺失和發生停機。

```
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
```

以下是執行此命令後成功回應的範例。

```
{
    "ClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/SecondTLSTest/
abcdabcd-1234-abcd-1234-abcd123e8e8e",
    "ClusterName": "ExampleClusterName",
    "State": "CREATING"
}
```

若要測試 TLS 加密

1. 依照 the section called “步驟 3：建立用戶端機器” 中的指引建立用戶端機器。
2. 在用戶端機器上安裝 Apache Kafka。
3. 在此範例中，我們使用 JVM 信任存放區與 MSK 叢集通話。若要這樣做，請先在用戶端機器上建立名為 /tmp 的資料夾。然後，前往 Apache Kafka 安裝的 bin 資料夾，並執行以下命令。(您的 JVM 路徑可能不同。)

```
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
```

4. 同時仍然在用戶端機器上的 Apache Kafka 安裝的 bin 資料夾中，建立一個名為 client.properties 且具有以下內容的文字檔案。

```
security.protocol=SSL
ssl.truststore.location=/tmp/kafka.client.truststore.jks
```

5. 在已 AWS CLI 安裝的機器上執行下列命令，並以*### ARN ## Cluster* ARN。

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

成功的結果看起來如下。儲存此結果，因為您下一個步驟需要它。

```
{
    "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"
}
```

6. 執行下列命令，取代*BootstrapBrokerStringTls*為您在上一個步驟中取得的其中一個 Broker 端點。

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

7. 開啟新的命令視窗並連線至相同的用戶端機器。然後，執行下列命令來建立主控台取用者。

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBrokerStringTls --consumer.config client.properties --topic
 TLSTestTopic
```

8. 在生產者視窗中，輸入文字訊息後方跟著換行符號，然後在取用者視窗中尋找相同的訊息。Amazon MSK 會在傳輸中加密此訊息。

如需有關設定 Apache Kafka 用戶端以使用加密資料的詳細資訊，請參閱 設定 Kafka 用戶端。

Amazon MSK API 的身分驗證和授權

AWS Identity and Access Management (IAM) 可協助管理員安全地控制 AWS 資源存取權。 AWS 服務 IAM 管理員可控制哪些人員可進行身分驗證 (登入) 並獲得授權 (具有許可) 以使用 Amazon MSK 資源。您可以使用 IAM AWS 服務 ，無需額外付費。

本頁說明如何使用 IAM 控制誰可以在您的叢集上執行 Amazon MSK 操作。如需有關如何控制誰可以在叢集上執行 Apache Kafka 操作的資訊，請參閱 the section called “Apache Kafka API 的身分驗證和授權”。

主題

- Amazon MSK 如何搭配 IAM 運作
- Amazon MSK 身分型政策範例
- 使用 Amazon MSK 的服務連結角色
- AWS Amazon MSK 的受管政策
- 疑難排解 Amazon MSK 身分和存取

Amazon MSK 如何搭配 IAM 運作

在您使用 IAM 管理對 Amazon MSK 的存取權之前，您應該了解哪些 IAM 功能可以與 Amazon MSK 搭配使用。若要深入瞭解 Amazon MSK 和其他 AWS 服務如何與 IAM 搭配使用，請參閱 IAM 使用者指南中的與 IAM 搭配使用的AWS 服務。

主題

- Amazon MSK 身分型政策
- Amazon MSK 資源型政策
- AWS 受管理政策
- 基於 Amazon MSK 標籤的授權
- Amazon MSK IAM 角色

Amazon MSK 身分型政策

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。Amazon MSK 支援特定動作、資源和條件鍵。若要了解您在 JSON 政策中使用的所有元素，請參閱 IAM 使用者指南中的 JSON 政策元素參考。

動作

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼条件下可以對什麼資源執行哪些動作。

JSON 政策的 Action 元素描述您可以用來允許或拒絕政策中存取的動作。原則動作通常與關聯的 AWS API 作業具有相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

Amazon MSK 中的政策動作會在動作之前使用下列前綴：kafka:。例如，若要授予某人使用 Amazon MSK DescribeCluster API 操作描述 MSK 叢集的許可，請在其政策中包含 kafka:DescribeCluster 動作。政策陳述式必須包含 Action 或 NotAction 元素。Amazon MSK 會定義自己的一組動作，描述您可以使用此服務執行的任務。

若要在單一陳述式中指定多個動作，請用逗號分隔，如下所示：

```
"Action": ["kafka:action1", "kafka:action2"]
```

您也可以使用萬用字元 (*) 來指定多個動作。例如，若要指定開頭是 Describe 文字的所有動作，請包含以下動作：

```
"Action": "kafka:Describe*"
```

若要查看 Amazon MSK 動作的清單，請參閱《IAM 使用者指南》中的 Amazon Managed Streaming for Apache Kafka 的動作、資源和條件鍵。

資源

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Resource JSON 政策元素可指定要套用動作的物件。陳述式必須包含 Resource 或 NotResource 元素。最佳實務是使用其 Amazon Resource Name (ARN) 來指定資源。您可以針對支援特定資源類型的動作 (稱為資源層級許可) 來這麼做。

對於不支援資源層級許可的動作 (例如列出操作)，請使用萬用字元 (*) 來表示陳述式適用於所有資源。

```
"Resource": "*"
```

Amazon MSK 執行個體資源具有以下 ARN：

```
arn:${Partition}:kafka:${Region}:${Account}:cluster/${ClusterName}/${UUID}
```

如需 ARN 格式的詳細資訊，請參閱 Amazon 資源名稱 (ARN) 和 AWS 服務命名空間。

例如，若要在陳述式中指定 CustomerMessages 執行個體，請使用以下 ARN：

```
"Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/CustomerMessages/abcd1234-abcd-dcba-4321-a1b2abcd9f9f-2"
```

若要指定屬於特定帳戶的所有執行個體，請使用萬用字元 (*)：

```
"Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/*"
```

有些 Amazon MSK 動作無法對特定資源執行，例如用來建立資源的動作。在這些情況下，您必須使用萬用字元 (*)。

```
"Resource": "*"
```

若要在單一陳述式中指定多項資源，請使用逗號分隔 ARN。

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

若要查看 Amazon MSK 資源類型及其 ARN 的清單，請參閱《IAM 使用者指南》中的 Amazon Managed Streaming for Apache Kafka 定義的資源。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱 Amazon Managed Streaming for Apache Kafka 定義的動作。

條件索引鍵

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Condition 元素 (或 Condition 區塊) 可讓您指定使陳述式生效的條件。Condition 元素是選用項目。您可以建立使用條件運算子的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。

若您在陳述式中指定多個 Condition 元素，或是在單一 Condition 元素中指定多個索引鍵， AWS 會使用邏輯 AND 操作評估他們。如果您為單一條件索引鍵指定多個值，請使用邏輯OR運算來 AWS 評估條件。必須符合所有條件，才會授與陳述式的許可。

您也可以在指定條件時使用預留位置變數。例如，您可以只在使用者使用其 IAM 使用者名稱標記時，將存取資源的許可授予該 IAM 使用者。如需更多資訊，請參閱 IAM 使用者指南中的 IAM 政策元素：變數和標籤。

AWS 支援全域條件金鑰和服務特定條件金鑰。若要查看所有 AWS 全域條件金鑰，請參閱《IAM 使用者指南》中的AWS 全域條件內容金鑰。

Amazon MSK 會定義自己的一組條件鍵，也支援使用一些全域條件鍵。若要查看所有 AWS 全域條件金鑰，請參閱 IAM 使用者指南中的AWS 全域條件內容金鑰。

若要查看 Amazon MSK 條件鍵的清單，請參閱《IAM 使用者指南》中的 Amazon Managed Streaming for Apache Kafka 的條件鍵。若要了解您可以搭配哪些動作和資源使用條件鍵，請參閱 Amazon Managed Streaming for Apache Kafka 定義的動作。

範例

若要檢視 Amazon MSK 身分型政策的範例，請參閱 Amazon MSK 身分型政策範例。

Amazon MSK 資源型政策

Amazon MSK 支援將叢集政策 (也稱為資源型政策) 與 Amazon MSK 叢集搭配使用。您可以使用叢集政策定義哪些 IAM 主體具有跨帳戶許可，可以設定 Amazon MSK 叢集的私有連線。與 IAM 用戶端身分驗證搭配使用時，您還可以使用叢集政策為連線的用戶端精細定義 Kafka 資料平面的許可。

若要檢視如何設定叢集政策的範例，請參閱步驟 2：將叢集政策連接至 MSK 叢集。

AWS 受管理政策

基於 Amazon MSK 標籤的授權

您可以將標籤連接至 Amazon MSK 叢集。若要根據標籤控制存取，請使用 `kafka:ResourceTag/`*`key-name`*、`aws:RequestTag/`*`key-name`* 或 `aws:TagKeys` 條件鍵，在政策的條件元素中，提供標籤資訊。如需有關標記 Amazon MSK 資源的詳細資訊，請參閱the section called “標記叢集”。

若要檢視身分型政策範例，了解如何根據標籤上的叢集來限制該叢集的存取權，請參閱 根據標籤存取 Amazon MSK 叢集。

Amazon MSK IAM 角色

IAM 角色是您 Amazon Web Services 帳戶中具備特定許可的實體。

將暫時憑證與 Amazon MSK 搭配使用

您可以搭配聯合使用暫時憑證、擔任 IAM 角色，或是擔任跨帳戶角色。您可以透過呼叫AssumeRole或GetFederation權杖等 AWS STS API 作業來取得臨時安全登入資料。

Amazon MSK 支援使用暫時憑證。

服務連結角色

服務連結角色允許 Amazon Web Services 存取其他服務中的資源，以代您完成動作。服務連結角色會顯示在您的 IAM 帳戶中，並由該服務所擁有。 管理員可以檢視，但不能編輯服務連結角色的許可。

Amazon MSK 支援服務連結角色。如需有關建立或管理 Amazon MSK 服務連結角色的詳細資訊，請參閱the section called “服務連結角色”。

Amazon MSK 身分型政策範例

根據預設，IAM 使用者和角色沒有執行 Amazon MSK API 動作的許可。管理員必須建立 IAM 政策，授與使用者和角色在指定資源上執行特定 API 操作所需的許可。管理員接著必須將這些政策連接至需要這些許可的 IAM 使用者或群組。

若要了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的在 JSON 標籤上建立政策。

主題

- 政策最佳實務
- 允許使用者檢視他們自己的許可
- 存取 Amazon MSK 叢集
- 根據標籤存取 Amazon MSK 叢集

政策最佳實務

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 Amazon MSK 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管原則並邁向最低權限權限 — 若要開始授與使用者和工作負載的權限，請使用可授與許多常見使用案例權限的AWS 受管理原則。它們可用在您的 AWS 帳戶. 建議您透過定義特定於您使用案例的 AWS 客戶管理政策，進一步降低使用權限。如需更多資訊，請參閱 IAM 使用者指南中的 AWS 受管政策或任務職能的AWS 受管政策。
- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱 IAM 使用者指南中的 IAM 中的政策和許可。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。您也可以使用條件來授與對服務動作的存取權 (如透過特定) 使用這些動作 AWS 服務，例如 AWS CloudFormation。如需詳細資訊，請參閱 IAM 使用者指南中的 IAM JSON 政策元素：條件。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您編寫安全且實用的政策。如需更多資訊，請參閱 IAM 使用者指南中的 IAM Access Analyzer 政策驗證。
- 需要多因素身份驗證 (MFA) — 如果您的案例需要 IAM 使用者或根使用者 AWS 帳戶，請開啟 MFA 以獲得額外的安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需更多資訊，請參閱 IAM 使用者指南中的設定 MFA 保護的 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱 IAM 使用者指南中的 IAM 安全最佳實務。

允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視附加到他們使用者身分的內嵌及受管政策。此原則包含在主控台上或以程式設計方式使用 AWS CLI 或 AWS API 完成此動作的權限。

```
{
    "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": "*"
        }
    ]
}
```

存取 Amazon MSK 叢集

在此範例中，您想要授予您 Amazon Web Services 帳戶中的 IAM 使用者存取您的其中一個叢集 `purchaseQueriesCluster`。此政策允許使用者描述叢集，獲取其引導代理程式，列出其代理程式節點並將其更新。

```
{
   "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"
      }
   ]
```

```
}
```

根據標籤存取 Amazon MSK 叢集

您可以在身分型政策中使用條件，根據標籤控制對 Amazon MSK 資源的存取權。此範例顯示您可能會如何建立政策，以允許使用者描述叢集、取得其引導代理程式、列出其代理節點、更新及刪除。但是，只有在叢集標籤 Owner 的值是該使用者的使用者名稱時，才會授予許可。

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

您可以將此政策連接到您帳戶中的 IAM 使用者。若名為 richard-roe 的使用者嘗試更新 MSK 叢集，則叢集必須被標記為 Owner=richard-roe 或 owner=richard-roe。否則，他便會被拒絕存取。條件標籤鍵 Owner 符合 Owner 和 owner，因為條件索引鍵名稱不區分大小寫。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素：條件。

使用 Amazon MSK 的服務連結角色

Amazon MSK 使用 AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是直接連結至 Amazon MSK 的一種特殊 IAM 角色類型。服務連結角色由 Amazon MSK 預先定義，並包含服務代表您呼叫其他 AWS 服務所需的所有許可。

服務連結角色可讓設定 Amazon MSK 更為簡單，因為您不必手動新增必要的許可。Amazon MSK 會定義其服務連結角色的許可。除非另有定義，否則僅 Amazon MSK 能擔任其角色。定義的許可包括信任政策和許可政策，且該許可政策無法附加至其他 IAM 實體。

如需有關支援服務連結角色的其他服務的資訊，請參閱可搭配 IAM 運作的 Amazon Web Services，並尋找服務連結角色資料欄顯示為是的服務。選擇具有連結的是，以檢視該服務的服務連結角色文件。

主題

- Amazon MSK 的服務連結角色許可
- 建立 Amazon MSK 的服務連結角色
- 編輯 Amazon MSK 的服務連結角色
- Amazon MSK 服務連結角色支援的區域

Amazon MSK 的服務連結角色許可

Amazon MSK 會使用名為 AWSServiceRoleForKafka 的服務連結角色。Amazon MSK 會使用此角色存取您的資源並執行以下操作：

- `*NetworkInterface`：在客戶帳戶中建立和管理網路介面，讓客戶 VPC 中的用戶端可以存取叢集代理程式。
- `*VpcEndpoints`— 在客戶帳戶中管理 VPC 端點，使客戶 VPC 中的用戶端可以使用叢集代理程式存取。 AWS PrivateLink Amazon MSK 會使用 `DescribeVpcEndpoints`、`ModifyVpcEndpoint` 和 `DeleteVpcEndpoints` 許可。
- `secretsmanager`— 管理客戶端憑據 AWS Secrets Manager.
- `GetCertificateAuthorityCertificate`：擷取私有憑證授權機構的憑證。

此服務連結角色連接至下列受管政策：`KafkaServiceRolePolicy`。如需此原則的更新，請參閱KafkaServiceRolePolicy。

AWSServiceRoleForKafka 服務連結角色信任下列服務以擔任角色：

- `kafka.amazonaws.com`

角色許可政策允許 Amazon MSK 對資源完成下列動作。

```
{
 "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_*"
    }
   }
  }
 ]
}
```

您必須設定許可，IAM 實體 (如使用者、群組或角色) 才可建立、編輯或刪除服務連結角色。如需詳細資訊，請參閱《IAM 使用者指南》中的服務連結角色許可。

建立 Amazon MSK 的服務連結角色

您不需要手動建立一個服務連結角色。當您在 AWS Management Console、或 AWS API 中建立 Amazon MSK 叢集時 AWS CLI，Amazon MSK 會為您建立服務連結角色。

若您刪除此服務連結角色，之後需要再次建立，您可以在帳戶中使用相同程序重新建立角色。當您建立 Amazon MSK 叢集時，Amazon MSK 會再次為您建立服務連結角色。

編輯 Amazon MSK 的服務連結角色

Amazon MSK 不允許您編輯 AWSServiceRoleForKafka 服務連結角色。因為有各種實體可能會參考服務連結角色，所以您無法在建立角色之後變更角色名稱。然而，您可使用 IAM 來編輯角色描述。如需更多資訊，請參閱 IAM 使用者指南中的編輯服務連結角色。

Amazon MSK 服務連結角色支援的區域

Amazon MSK 在所有提供此服務的區域中支援使用服務連結的角色。如需詳細資訊，請參閱 AWS 區域與端點。

AWS Amazon MSK 的受管政策

受 AWS 管理的策略是由建立和管理的獨立策略 AWS。 AWS 受管理的策略旨在為許多常見使用案例提供權限，以便您可以開始將權限指派給使用者、群組和角色。

請記住， AWS 受管理的政策可能不會為您的特定使用案例授與最低權限權限，因為這些權限可供所有 AWS 客戶使用。我們建議您定義使用案例專屬的客戶管理政策，以便進一步減少許可。

您無法變更受 AWS 管理策略中定義的權限。如果 AWS 更新 AWS 受管理原則中定義的權限，則此更新會影響附加原則的所有主體識別 (使用者、群組和角色)。 AWS 當新的啟動或新 AWS 服務 的 API 操作可用於現有服務時，最有可能更新 AWS 受管理策略。

如需詳細資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策。

AWS 管理策略：亞馬遜 FullAccess

此政策會授予管理許可，允許主體完整存取所有 Amazon MSK 動作。此政策中的許可分組如下：

- Amazon MSK 許可允許所有 Amazon MSK 動作。
- **Amazon EC2權限** — 在此策略中需要驗證 API 請求中傳遞的資源。這是為了確保 Amazon MSK 能夠成功使用叢集資源。此政策中的其餘 Amazon EC2 許可允許 Amazon MSK 建立所需的 AWS 資源，讓您能夠連線到叢集。
- **AWS KMS權限**-在 API 調用期間用於驗證請求中傳遞的資源。Amazon MSK 需要它們才能使用 Amazon MSK 叢集傳遞的密鑰。
- **CloudWatch Logs, Amazon S3, and Amazon Data Firehose權限** — Amazon MSK 必須要有權限才能確保可連線到日誌傳遞目的地，並且對於代理程式日誌使用有效。
- **IAM許可** — Amazon MSK 必須能夠在您的帳戶中建立服務連結角色，並允許您將服務執行角色傳遞給 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 管理策略：亞馬遜 MSK 訪ReadOnly問

此政策授予唯讀許可，允許使用者檢視 Amazon MSK 中的資訊。連接此政策的主體無法進行任何更新或刪除現有資源，也無法建立新的 Amazon MSK 資源。例如，具有這些許可的主體可以檢視與其帳戶相關聯的叢集和組態清單，但無法變更任何叢集的組態或設定。此政策中的許可分組如下：

- **Amazon MSK**許可 — 允許您列出 Amazon MSK 資源、描述資源並取得有關它們的資訊。
- **Amazon EC2**許可 — 用於描述與叢集關聯的 Amazon VPC、子網路、安全群組和 ENI。
- **AWS KMS**權限-用於描述與集群相關聯的密鑰。

```
{
    "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 受管理的策略： KafkaServiceRolePolicy

您無法附加 KafkaServiceRolePolicy 到 IAM 實體。此政策會連接到服務連結角色，可讓 Amazon MSK 執行動作，例如管理 MSK 叢集上的 VPC 端點 (連接器)、管理網路介面，以及透過 AWS Secrets Manager管理叢集憑證。如需詳細資訊，請參閱 the section called “服務連結角色”。

AWS 受管理的策略： AWSMSKReplicatorExecutionRole

該AWSMSKReplicatorExecutionRole政策授予 Amazon MSK 複寫器的許可，以便在 MSK 叢集之間複寫資料。此政策中的許可分組如下：

- **cluster**— 授予 Amazon MSK 複製器許可，以使用 IAM 身分驗證連接到叢集。也會授與描述和變更叢集的權限。
- **topic**— 授予 Amazon MSK 複製器許可，以描述、建立和更改主題，以及變更主題的動態組態。
- **consumer group**— 授與 Amazon MSK 複製器權限，以描述和更改用戶群組、從 MSK 叢集讀取和寫入日期，以及刪除複寫器建立的內部主題。

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "ClusterPermissions",
   "Effect": "Allow",
   "Action": [
    "kafka-cluster:Connect",
    "kafka-cluster:DescribeCluster",
    "kafka-cluster:AlterCluster",
    "kafka-cluster:DescribeTopic",
    "kafka-cluster:CreateTopic",
    "kafka-cluster:AlterTopic",
    "kafka-cluster:WriteData",
```

```
    "kafka-cluster:ReadData",
    "kafka-cluster:AlterGroup",
    "kafka-cluster:DescribeGroup",
    "kafka-cluster:DescribeTopicDynamicConfiguration",
    "kafka-cluster:AlterTopicDynamicConfiguration",
    "kafka-cluster:WriteDataIdempotently"
   ],
   "Resource": [
    "arn:aws:kafka:*:*:cluster/*"
   ]
  },
  {
   "Sid": "TopicPermissions",
   "Effect": "Allow",
   "Action": [
    "kafka-cluster:DescribeTopic",
    "kafka-cluster:CreateTopic",
    "kafka-cluster:AlterTopic",
    "kafka-cluster:WriteData",
    "kafka-cluster:ReadData",
    "kafka-cluster:DescribeTopicDynamicConfiguration",
    "kafka-cluster:AlterTopicDynamicConfiguration",
    "kafka-cluster:AlterCluster"
   ],
   "Resource": [
    "arn:aws:kafka:*:*:topic/*/*"
   ]
  },
  {
   "Sid": "GroupPermissions",
   "Effect": "Allow",
   "Action": [
    "kafka-cluster:AlterGroup",
    "kafka-cluster:DescribeGroup"
   ],
   "Resource": [
    "arn:aws:kafka:*:*:group/*/*"
   ]
  }
 ]
}
```

Amazon MSK 更新 AWS 受管政策

檢視 Amazon MSK AWS 受管政策更新的詳細資訊，因為此服務開始追蹤這些變更。

變更	描述	日期
WriteDataIdempotently 權限新增至 AWSMSKReplicatorExecutionRole — 更新至現有策略	Amazon MSK 新增 AWSMSKReplicatorExecutionRole 政策 WriteDataIdempotently 許可，以支援 MSK 叢集之間的資料複寫。	2024年3月12日
AWSMSKReplicatorExecutionRole – 新政策	Amazon MSK 增加了 AWSMSKReplicatorExecutionRole 政策以支持 Amazon MSK 複製器。	2023 年 12 月 4 日
亞馬遜 MSK FullAccess — 更新到現有政策	Amazon MSK 新增許可以支援 Amazon MSK Replicator。	2023 年 9 月 28 日
KafkaServiceRolePolicy – 更新現有政策	Amazon MSK 新增許可以支援多 VPC 私有連線。	2023 年 3 月 8 日
亞馬遜 MSK FullAccess — 更新到現有政策	Amazon MSK 新增 Amazon EC2 許可，可以連線到叢集。	2021 年 11 月 30 日
亞馬遜 MSK FullAccess — 更新到現有政策	Amazon MSK 新增許可，允許描述 Amazon EC2 路由表。	2021 年 11 月 19 日
Amazon MSK 開始追蹤變更	Amazon MSK 開始追蹤其 AWS 受管政策的變更。	2021 年 11 月 19 日

疑難排解 Amazon MSK 身分和存取

請使用以下資訊來協助您診斷和修正使用 Amazon MSK 和 IAM 時可能遇到的常見問題。

主題

- 我未獲授權，不得在 Amazon MSK 中執行動作

我未獲授權，不得在 Amazon MSK 中執行動作

如果 AWS Management Console 告訴您您沒有執行動作的授權，則您必須聯絡您的管理員以尋求協助。您的管理員是為您提供簽署憑證的人員。

以下範例錯誤會在 `mateojackson` IAM 使用者嘗試使用主控台刪除叢集，但卻沒有 `kafka:DeleteCluster` 許可時發生。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 kafka:DeleteCluster on resource: purchaseQueriesCluster
```

在此情況下，Mateo 會請求管理員更新他的政策，允許他使用 `purchaseQueriesCluster` 動作存取 `kafka:DeleteCluster` 資源。

Apache Kafka API 的身分驗證和授權

您可以使用 IAM 來對用戶端進行身分驗證，以及允許或拒絕 Apache Kafka 動作。也可以使用 TLS 或 SASL/SCRAM 來來對用戶端進行身分驗證，以及使用 Apache Kafka ACL 來允許或拒絕動作。

如需有關如何控制誰可以在叢集上執行 Amazon MSK 操作的相關資訊，請參閱 the section called “Amazon MSK API 的身分驗證和授權”。

主題

- IAM 存取控制
- 交互 TLS 驗證
- 使用 AWS Secrets Manager 登入認證驗證
- Apache Kafka ACL

IAM 存取控制

適用於 Amazon MSK 的 IAM 存取控制可讓您處理 MSK 叢集的身分驗證和授權。這樣，您就對身分驗證和授權分別使用不同的機制。例如，當用戶端嘗試寫入至您的叢集時，Amazon MSK 會使用 IAM 來檢查該用戶端是否為已驗證的身分，以及是否已獲授權可對您的叢集進行生產。IAM 存取控制適用於 Java 和非 Java 用戶端，包括使用 Python、Go 和 .NET 編寫的卡夫卡用戶端。 JavaScript

Amazon MSK 會記錄存取事件以便您稽核。如需詳細資訊，請參閱 the section called “CloudTrail 事件”。

為了讓 IAM 存取控制得以運作，Amazon MSK 會對 Apache Kafka 的原始程式碼進行輕微修改。這些修改不會對您的 Apache Kafka 使用產生明顯差異。

> **Important**
>
> IAM 存取控制不適用於 Apache ZooKeeper 節點。如需有關如何控制節點存取權的詳細資訊，請參閱the section called “控制對阿帕奇的存取 ZooKeeper”。

> **Important**
>
> 如果您的叢集使用 IAM 存取控制，則 `allow.everyone.if.no.acl.found` Apache Kafka 設定將無作用。

> **Important**
>
> 您可以針對使用 IAM 存取控制的 MSK 叢集調用 Apache Kafka ACL API。但是，阿帕奇卡夫卡 ACL 對 IAM 角色的授權沒有影響。您必須使用 IAM 政策對 IAM 角色進行存取控制。

Amazon MSK 的 IAM 存取控制運作方式

若要針對 Amazon MSK 使用 IAM 存取控制，請執行以下步驟，本節的其餘部分將有詳細說明。

- the section called “建立使用 IAM 存取控制的叢集”
- the section called “設定 IAM 存取控制的用戶端”
- the section called “建立授權政策”
- the section called “取得 IAM 存取控制的引導代理程式”

建立使用 IAM 存取控制的叢集

本節說明如何使用 AWS Management Console、API 或建立使 AWS CLI 用 IAM 存取控制的叢集。如需有關如何啟用現有叢集的 IAM 存取控制的資訊，請參閱the section called “更新安全性”。

使用建 AWS Management Console 立使用 IAM 存取控制的叢集

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。

2. 選擇建立叢集。
3. 選擇使用自訂設定建立叢集。
4. 在身分驗證區段中，選擇 IAM 存取控制。
5. 完成其餘工作流程以建立叢集。

使用 API 或建立 AWS CLI 使用 IAM 存取控制的叢集

- 若要建立已啟用 IAM 存取控制的叢集，請使用 CreateClusterAPI 或建立叢集 CLI 命令，並為參數傳遞下列 JSON：`ClientAuthentication"ClientAuthentication": { "Sasl": { "Iam": { "Enabled": true } }`

設定 IAM 存取控制的用戶端

若要讓用戶端能夠與使用 IAM 存取控制的 MSK 叢集通訊，您可以使用下列其中一種機制：

- 使用 SASL_OAUTHBEARER 機制的非 Java 用戶端組態
- 使用 SASL_OAUTHBEARER 機制或 AWS_MSK_IAM 機制的 Java 用戶端組態

使用 SASL_OAUTHBEARER 機制來設定 IAM

1. 編輯您的 client.properties 組態檔案，並使用下面的 Python Kafka 範例用戶端中反白的語法作為指南。其他語言的組態變更也與其類似。

```
#!/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. 下載您選擇的組態語言的協助程式庫，並依照該語言程式庫首頁入門區段中的指示操作。
   - JavaScript: https://github.com/aws/aws-msk-iam-sasl-signer-js#getting-started
   - Python：https://github.com/aws/aws-msk-iam-sasl-signer-python#get-started
   - Go：https://github.com/aws/aws-msk-iam-sasl-signer-go#getting-started
   - .NET：https://github.com/aws/aws-msk-iam-sasl-signer-net#getting-started
   - JAVA：可透過 `aws-msk-iam-auth` jar 檔案獲得 Java 的 SASL_OAUTHBEARE 支援

使用 MSK 自訂 AWS_MSK_IAM 機制來設定 IAM

1. 將以下內容新增到 `client.properties` 檔案。使用用戶端信任存放區檔案的完整路徑取代 *<PATH_TO_TRUST_STORE_FILE>*。

   > **Note**
   >
   > 如果您不想使用特定憑證，可以從 `client.properties` 檔案中移除 `ssl.truststore.location=`*<PATH_TO_TRUST_STORE_FILE>*。如果您沒有為 `ssl.truststore.location` 指定值，Java 程序會使用預設憑證。

   ```
   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
```

若要使用您為 AWS 認證建立的具名設定檔，請包含`awsProfileName="your profile name";`在用戶端組態檔中。若要取得有關具名紀要的資訊，請參閱 AWS CLI 文件中的具名紀要。

2. 下載最新的穩定版 aws-msk-iam-auth JAR 文件，並將其放置在類路徑中。如果您使用 Maven，請新增以下依賴項，根據需要調整版本號：

```
<dependency>
    <groupId>software.amazon.msk</groupId>
    <artifactId>aws-msk-iam-auth</artifactId>
    <version>1.0.0</version>
</dependency>
```

Amazon MSK 用戶端外掛程式已在 Apache 2.0 授權下開放原始碼。

建立授權政策

將授權政策連接至對應用戶端的 IAM 角色。在授權政策中，您可以指定角色要允許或拒絕的動作。如果您的用戶端位於 Amazon EC2 執行個體上，請將授權政策與該 Amazon EC2 執行個體的 IAM 角色建立關聯。或者，您可以將用戶端設定為使用命名設定檔，然後將授權政策與該命名設定檔的角色建立關聯。the section called “設定 IAM 存取控制的用戶端” 說明如何設定用戶端以使用命名設定檔。

如需有關建立 IAM 政策的詳細資訊，請參閱建立 IAM 政策。

以下是名為的叢集的授權原則範例 MyTestCluster。若要了解 `Action` 和 `Resource` 元素的語意，請參閱the section called “動作和資源的語義”。

> **Important**
>
> 您對 IAM 政策所做的變更會立即反映在 IAM API 與 AWS CLI 中。不過，政策變更可能需要一點時間才會生效。在大多數情況下，政策變更會在一分鐘內生效。有時網路狀況可能會增加延遲。

```
{
    "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/*"
            ]
        }
    ]
}
```

若要了解如何使用對應常見 Apache Kafka 使用案例的動作元素 (例如產生和使用資料) 建立政策，請參閱the section called “常用案例”。

對於卡夫卡 2.8.0 及更高版本，WriteData不推薦使用等級權限（KIP-679）。預設為 `enable.idempotence = true`。因此，對於 Kafka 2.8.0 及以上版本，IAM 不提供與 Kafka ACL 相同的功能。如果僅提供對主題的 `WriteData` 存取權，就不可能對該主題進行

WriteDataIdempotently。這不會影響 WriteData 被提供給所有主題的情況。在這種情況下，WriteDataIdempotently 是允許的。這是因為 IAM 邏輯的實作方式與 Kafka ACL 有所不同。

若要解決這個問題，我們建議您使用類似以下範例的政策：

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

在這種情況下，WriteData 允許寫入 TestTopic，同時 WriteDataIdempotently 允許等冪寫入叢集。需要注意，WriteDataIdempotently 是一個叢集層級許可。它不能在主題層級使用。如果 WriteDataIdempotently 僅限於主題層級，則此政策將無法運作。

取得 IAM 存取控制的引導代理程式

請參閱the section called “取得引導代理程式”。

動作和資源的語義

本節說明您可以在 IAM 授權政策中使用的動作和資源元素的語義。如需政策範例，請參閱 the section called “建立授權政策”。

動作

下表列出在 Amazon MSK 使用 IAM 存取控制時，可包含在授權政策中的動作。當您在授權政策中加入來自表格動作資料欄中的動作時，您還必須加入必要動作資料欄中的對應動作。

動作	描述	必要的動作	必要的資源	適用於無伺服器叢集
`kafka-cluster:Connect`	准許與叢集連線並進行身分驗證。	無	叢集	是
`kafka-cluster:DescribeCluster`	准許描述叢集的各方面，相當於 Apache Kafka 的 DESCRIBE CLUSTER ACL。	`kafka-cluster:Connect`	叢集	是

`kafka-cluster:AlterCluster`	准許改變叢集的各方面，相當於 Apache Kafka 的 ALTER CLUSTER ACL。	`kafka-cluster:Connect`
		
		`kafka-cluster:DescribeCluster`
			叢集	否

`kafka-cluster:DescribeClusterDynamicConfiguration`	准許描述叢集的動態組態，相當於 Apache Kafka 的 DESCRIBE_CONFIGS	`kafka-cluster:Connect`	叢集	否

動作	描述	必要的動作	必要的資源	適用於無伺服器叢集
	CLUSTER ACL。			
kafka-cluster:AlterClusterDynamicConfiguration	准許改變叢集的動態組態，相當於 Apache Kafka 的 ALTER_CONFIGS CLUSTER ACL。	kafka-cluster:Connect kafka-cluster:DescribeClusterDynamicConfiguration	叢集	否
kafka-cluster:WriteDataIdempotently	准許在叢集上等冪寫入資料，相當於 Apache Kafka 的 IDEMPOTENT_WRITE CLUSTER ACL。	kafka-cluster:Connect kafka-cluster:WriteData	叢集	是
kafka-cluster:CreateTopic	准許在叢集上建立主題，相當於 Apache Kafka 的 CREATE CLUSTER/TOPIC ACL。	kafka-cluster:Connect	主題	是
kafka-cluster:DescribeTopic	准許描述叢集上的主題，相當於 Apache Kafka 的 DESCRIBE TOPIC ACL。	kafka-cluster:Connect	主題	是

動作	描述	必要的動作	必要的資源	適用於無伺服器叢集
kafka-cluster:AlterTopic	准許改變叢集上的主題，相當於 Apache Kafka 的 ALTER TOPIC ACL。	kafka-cluster:Connect kafka-cluster:DescribeTopic	主題	是
kafka-cluster:DeleteTopic	准許刪除叢集上的主題，相當於 Apache Kafka 的 DELETE TOPIC ACL。	kafka-cluster:Connect kafka-cluster:DescribeTopic	主題	是
kafka-cluster:DescribeTopicDynamicConfiguration	准許描述叢集上主題的動態組態，相當於 Apache Kafka 的 DESCRIBE_CONFIGS TOPIC ACL。	kafka-cluster:Connect	主題	是
kafka-cluster:AlterTopicDynamicConfiguration	准許改變叢集上主題的動態組態，相當於 Apache Kafka 的 ALTER_CONFIGS TOPIC ACL。	kafka-cluster:Connect kafka-cluster:DescribeTopicDynamicConfiguration	主題	是

動作	描述	必要的動作	必要的資源	適用於無伺服器叢集
kafka-cluster:ReadData	准許讀取叢集上主題的資料，相當於 Apache Kafka 的 READ TOPIC ACL。	kafka-cluster:Connect kafka-cluster:DescribeTopic kafka-cluster:AlterGroup	主題	是
kafka-cluster:WriteData	准許寫入叢集上主題的資料，相當於 Apache Kafka 的 WRITE TOPIC ACL。	kafka-cluster:Connect kafka-cluster:DescribeTopic	主題	是
kafka-cluster:DescribeGroup	准許描述叢集上的群組，相當於 Apache Kafka 的 DESCRIBE GROUP ACL。	kafka-cluster:Connect	群組	是
kafka-cluster:AlterGroup	准許加入叢集上的群組，相當於 Apache Kafka 的 READ GROUP ACL。	kafka-cluster:Connect kafka-cluster:DescribeGroup	群組	是

動作	描述	必要的動作	必要的資源	適用於無伺服器叢集
kafka-cluster:DeleteGroup	准許刪除叢集上的群組，相當於 Apache Kafka 的 DELETE GROUP ACL。	kafka-cluster:Connect kafka-cluster:DescribeGroup	群組	是
kafka-cluster:DescribeTransactionalId	准許描述叢集上的交易 ID，相當於 Apache Kafka 的 DESCRIBE TRANSACTIONAL_ID ACL。	kafka-cluster:Connect	transactional-id	是
kafka-cluster:AlterTransactionalId	准許改變叢集上的交易 ID，相當於 Apache Kafka 的 WRITE TRANSACTIONAL_ID ACL。	kafka-cluster:Connect kafka-cluster:DescribeTransactionalId kafka-cluster:WriteData	transactional-id	是

您可以在動作中的冒號後面使用星號 (*) 萬用字元任意次數。範例如下。

- 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:*` 代表所有許可。

資源

下表顯示將 IAM 存取控制用於 Amazon MSK 時，可在授權政策中使用的四種資源類型。您可以從或使用 DescribeClusterAPI AWS Management Console 或描述 AWS CLI 叢集命令取得叢集 Amazon 資源名稱 (ARN)。然後，您可以使用叢集 ARN 來建構主題、群組和交易 ID ARN。若要在授權政策中指定資源，請使用該資源的 ARN。

資源	ARN 格式
叢集	arn:aws:kafka:*region*:*account-id* :cluster/*cluster-name* /*cluster-uuid*
主題	arn:aws:kafka:*region*:*account-id* :topic/*cluster-name* /*cluster-uuid* /*topic-name*
群組	arn:aws:kafka:*region*:*account-id* :group/*cluster-name* /*cluster-uuid* /*group-name*
交易 ID	arn:aws:kafka:*region*:*account-id* :transactional-id/*cluster-name* /*cluster-uuid* /*transactional-id*

您可以在 ARN 中的 `:cluster/`、`:topic/`、`:group/`、`:transactional-id/` 後隨時使用星號 (*) 萬用字元任意次數。以下是使用星號 (*) 萬用字元表示多種資源的範例：

- `arn:aws:kafka:us-east-1:0123456789012:topic/MyTestCluster/*`：任何名為的叢集中的所有主題 MyTestCluster，不論叢集的 UUID 為何。
- `arn:aws:kafka:us-east-1:0123456789012:topic/MyTestCluster/abcd1234-0123-abcd-5678-1234abcd-1/*_test`：在叢集中名稱以「_test」結尾的所有主題，其名稱是，MyTestCluster 且其 UUID 為 ABCD-5678-1234ACD-1。
- `arn:aws:kafka:us-east-1:0123456789012:transactional-id/MyTestCluster/*/5555abcd-1111-abcd-1234-abcd1234-1`：其交易識別碼為 5555abcd-1111-ABCD-1234-ABCD1234-1 的所有交易，在您帳戶中指定的叢集的所有化身。 MyTestCluster 這表示，如果您建立名為的叢集 MyTestCluster，然後將其刪除，然後以相同名稱建立另一個叢集，則可以使用此資源 ARN 代表兩個叢集上的相同交易 ID。但是，您無法存取已刪除的叢集。

常用案例

下表中的第一個資料欄顯示一些常見的使用案例。若要授權用戶端執行指定的使用案例，請在用戶端的授權政策中包含該使用案例所需的動作，然後將 `Effect` 設定為 `Allow`。

如需有關 Amazon MSK 上所有 IAM 存取控制動作的相關資訊，請參閱the section called “動作和資源的語義”。

> **Note**
>
> 根據預設，動作會被拒絕。您必須明確允許要授權用戶端執行的每個動作。

使用案例	必要的動作
管理員	`kafka-cluster:*`
建立主題	`kafka-cluster:Connect` `kafka-cluster:CreateTopic`

產生資料	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:WriteData`

取用資料	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:DescribeGroup`
	`kafka-cluster:AlterGroup`
	`kafka-cluster:ReadData`

等冪產生資料	`kafka-cluster:Connect`
	`kafka-cluster:DescribeTopic`
	`kafka-cluster:WriteData`

使用案例	必要的動作
	kafka-cluster:WriteDataIdempotently
交易產生資料	kafka-cluster:Connect kafka-cluster:DescribeTopic kafka-cluster:WriteData kafka-cluster:DescribeTransactionalId kafka-cluster:AlterTransactionalId
說明叢集的組態	kafka-cluster:Connect kafka-cluster:DescribeClusterDynamicConfiguration
更新叢集的組態	kafka-cluster:Connect kafka-cluster:DescribeClusterDynamicConfiguration kafka-cluster:AlterClusterDynamicConfiguration
說明主題的組態	kafka-cluster:Connect kafka-cluster:DescribeTopicDynamicConfiguration

使用案例	必要的動作
更新主題的組態	kafka-cluster:Connect kafka-cluster:DescribeTopicDynamicConfiguration kafka-cluster:AlterTopicDynamicConfiguration
改變主題	kafka-cluster:Connect kafka-cluster:DescribeTopic kafka-cluster:AlterTopic

交互 TLS 驗證

您可以透過 TLS 啟用用戶端身份驗證，以便從應用程式到 Amazon MSK 代理程式的連線。如要使用用戶端身分驗證，您需要一個 AWS 私有 CA。 AWS 私有 CA 可以位於與您的叢集相 AWS 帳戶 同，也可以位於不同的帳戶中。如需有關 AWS 私有 CA的資訊，請參閱建立和管理 AWS 私有 CA.

> **Note**
>
> 北京和寧夏區域目前無法使用 TLS 身分驗證。

Amazon MSK 不支援憑證撤銷清單 (CRL)。若要控制對叢集主題的存取或封鎖遭到入侵的憑證，請使用 Apache Kafka ACL 和 AWS 安全性群組。如需有關使用 Apache Kafka ACL 的詳細資訊，請參閱 the section called “Apache Kafka ACL”。

本主題包含下列章節：

- 若要建立支援用戶端身分驗證的叢集
- 設定用戶端以使用身分驗證
- 使用身分驗證產生和使用訊息

若要建立支援用戶端身分驗證的叢集

此程序說明如何使用 AWS 私有 CA.

> **Note**
>
> 當您使用相互 TLS 來控制存取時，強烈建議您 AWS 私有 CA 為每個 MSK 叢集使用獨立的。這樣做可確保由 PCA 簽署的 TLS 憑證僅透過單一 MSK 叢集進行身分驗證。

1. 使用下列內容建立名為 `clientauthinfo.json` 的檔案。將 *Private-CA-ARN* 取代為您 PCA 的 ARN。

```
{
   "Tls": {
       "CertificateAuthorityArnList": ["Private-CA-ARN"]
    }
}
```

2. 建立名為 `brokernodegroupinfo.json` 的檔案，如 the section called “使用建立叢集 AWS CLI” 中所說明。
3. 用戶端身分驗證要求您也啟用用戶端和代理程式之間的傳輸中加密。使用下列內容建立名為 `encryptioninfo.json` 的檔案。將 *KMS-Key-ARN* 取代為您 KMS 金鑰的 ARN。您可以設定 `ClientBroker` 為 `TLS` 或 `TLS_PLAINTEXT`。

```
{
   "EncryptionAtRest": {
       "DataVolumeKMSKeyId": "KMS-Key-ARN"
    },
   "EncryptionInTransit": {
        "InCluster": true,
        "ClientBroker": "TLS"
    }
}
```

   如需加密的詳細資訊，請參閱 the section called “加密”。
4. 在已 AWS CLI 安裝的機器上，執行下列命令以建立啟用驗證和傳輸中加密的叢集。儲存回應中提供的叢集 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
```

設定用戶端以使用身分驗證

1. 建立用作用戶端機器的 Amazon EC2 執行個體。為求簡化，請在與叢集相同的 VPC 中建立此執行個體。如需如何建立這類用戶端機器的範例，請參閱 the section called “步驟 3：建立用戶端機器”。
2. 建立主題。如需範例，請參閱 the section called “步驟 4：建立主題” 下方的說明。
3. 在已 AWS CLI 安裝的機器上，執行下列命令以取得叢集的啟動程式代理程式。將 *Cluster-ARN* 取代為您的叢集 ARN。

   ```
   aws kafka get-bootstrap-brokers --cluster-arn Cluster-ARN
   ```

   儲存回應中與 BootstrapBrokerStringTls 相關聯的字串。
4. 在用戶端機器上，執行下列命令以使用 JVM 信任存放區來建立用戶端信任存放區。如果您的 JVM 路徑不同，請相應地調整命令。

   ```
   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. 在用戶端機器上，執行下列命令以建立用戶端私有金鑰。將 *Distinguished-Name*、*Example-Alias*、*Your-Store-Pass* 和 *Your-Key-Pass* 取代為您選擇的字串。

   ```
   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. 在用戶端機器上，執行下列命令，以使用您在上一個步驟中建立的私有金鑰來建立憑證要求。

   ```
   keytool -keystore kafka.client.keystore.jks -certreq -file client-cert-sign-request
    -alias Example-Alias -storepass Your-Store-Pass -keypass Your-Key-Pass
   ```
7. 打開 client-cert-sign-request 文件並確保以 -----BEGIN CERTIFICATE REQUEST----- 開始和以 -----END CERTIFICATE REQUEST----- 結束。如果開頭為

`-----BEGIN NEW CERTIFICATE REQUEST-----`，請從檔案的開頭和結尾刪除單字 NEW (以及其後的單一空格)。

8. 在已 AWS CLI 安裝的機器上，執行下列命令來簽署憑證要求。將 *Private-CA-ARN* 取代為您 PCA 的 ARN。如果想要，您可以變更有效性值。在這裡，我們使用 300 做為範例。

```
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"
```

儲存回應中提供的憑證 ARN。

> **Note**
>
> 若要擷取用戶端憑證，請使用 `acm-pca get-certificate` 指令並指定憑證 ARN。如需詳細資訊，請參閱 AWS CLI Command Reference 中的 get-certificate。

9. 執行下列命令以取得為您 AWS 私有 CA 簽署的憑證。將 *Certificate-ARN* 取代為您從先前命令的回應中取得的 ARN。

```
aws acm-pca get-certificate --certificate-authority-arn Private-CA-ARN --
certificate-arn Certificate-ARN
```

10. 從執行上一個命令的 JSON 結果中，複製與 `Certificate` 和 `CertificateChain` 相關聯的字串。將這兩個字符串粘貼到名為的新文件中 signed-certificate-from-acm。首先貼上與 `Certificate` 相關連的字串，接著與 `CertificateChain` 相關聯的字串。將 `\n` 取代為新行字元。以下是您貼上憑證和憑證鏈結之後的檔案結構。

```
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
```

11. 在用戶端機器上執行以下命令以將此憑證新增到您的金鑰存放區，以便其在與 MSK 代理程式通話時顯示。

```
keytool -keystore kafka.client.keystore.jks -import -file signed-certificate-from-
acm -alias Example-Alias -storepass Your-Store-Pass -keypass Your-Key-Pass
```

12. 使用下列內容建立名為 client.properties 的檔案。將信任庫和密鑰庫位置調整為您儲存 kafka.client.truststore.jks 的路徑。使用您的 Kafka 用戶端版本取代 *{YOUR KAFKA VERSION}* 預留位置。

```
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
```

使用身分驗證產生和使用訊息

1. 執行下列命令以建立主題。名為的檔案 client.properties 是您在上一個程序中建立的檔案。

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

2. 執行下列命令以啟動主控台生產者。名為的檔案 client.properties 是您在上一個程序中建立的檔案。

```
<path-to-your-kafka-installation>/bin/kafka-console-producer.sh --bootstrap-
server BootstrapBroker-String --topic ExampleTopic --producer.config
 client.properties
```

3. 在用戶端機器的新命令視窗中，執行下列命令以啟動主控台取用者。

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBroker-String --topic ExampleTopic --consumer.config
 client.properties
```

4. 在生產者視窗中輸入訊息，並觀看訊息出現在取用者視窗中。

使用 AWS Secrets Manager 登入認證驗證

您可以使用透過 AWS Secrets Manager 儲存和保護登入資料來控制對 Amazon MSK 叢集的存取。將使用者憑證儲存在 Secrets Manager 中，可減少叢集身分驗證 (例如稽核、更新和輪換憑證) 的額外負荷。Secrets Manager 也可讓您跨叢集共用使用者憑證。

本主題包含下列章節：

運作方式

Amazon MSK 的登入憑證身分驗證會使用 SASL/SCRAM (簡易身分驗證及安全性階層/Salted Challenge Response Mechanism) 身分驗證。若要設定叢集的登入憑證身分驗證，您可以在 AWS Secrets Manager 中建立秘密資源，並將登入憑證與該秘密建立關聯。

SAS/SCRM 定義見 RFC 5802。SCRAM 使用安全的雜湊演算法，不會在用戶端與伺服器之間傳輸純文字登入憑證。

> **Note**
>
> 為叢集設定 SASL/SCRAM 身分驗證後，Amazon MSK 會為用戶端和代理程式之間的所有流量開啟 TLS 加密。

為 Amazon MSK 叢集設定 SASL/SCRAM 身分驗證

若要在 Secret Manager 中設定密碼，請遵循 AWS Secrets Manager 使用者指南中的〈建立和擷取 AWS 密碼〉教學課程。

為 Amazon MSK 叢集建立秘密時，請注意以下要求：

- 針對秘密類型，選擇其他秘密類型 (如 API 金鑰)。
- 您的秘密名稱必須以 Amazon MSK_ 字首開頭。
- 您必須使用現有的自訂 AWS KMS 金鑰，或為密碼建立新的自訂 AWS KMS 金鑰。根據預設，Secrets Manager 會使用預設 AWS KMS 金鑰做為密碼。

> **Important**
>
> 使用預設 AWS KMS 金鑰建立的密碼無法與 Amazon MSK 叢集搭配使用。

- 您的登入憑證資料必須採用下列格式，才能使用純文字選項輸入鍵值對。

```
{
  "username": "alice",
  "password": "alice-secret"
}
```

- 記錄秘密的 ARN (Amazon Resource Name) 值。
-

> **Important**
>
> 您無法將 Secret Manager 秘密與超過 the section called “ 適當調整叢集大小：每個代理程式的分區數量” 中所述限制的叢集建立關聯。

- 如果您使用 AWS CLI 建立密碼，請指定`kms-key-id`參數的金鑰識別碼或 ARN。請勿指定別名。
- 若要將密碼與叢集建立關聯，請使用 Amazon MSK 主控台或 BatchAssociateScramSecret作業。

> **Important**
>
> 將秘密與叢集建立關聯後，Amazon MSK 會將資源政策連接至秘密，以便叢集能夠存取和讀取您定義的秘密值。您不應該修改此資源政策。這樣做可以防止您的叢集存取您的秘密。

下列範例中的 `BatchAssociateScramSecret` 操作 JSON 輸入會將秘密與叢集建立關聯：

```
{
  "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"
  ]
}
```

使用登入憑證連線至叢集

建立秘密並將其與叢集建立關聯後，即可將用戶端連線至叢集。下列範例步驟示範如何將用戶端連線到使用 SASL/SCRAM 身分驗證的叢集，以及如何產生資料到範例主題和取用範例主題的資料。

1. 在已安裝 AWS CLI 的機器上執行下列命令，並以*### ARN ####* ARN。

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

2. 若要建立範例主題，請執行下列命令，將 *BootstrapServerString* 取代為您在上一個步驟中取得的其中一個 Broker 端點。

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

3. 在用戶端機器上建立 JAAS 組態檔案，其中內含儲存在秘密中的使用者憑證。例如，為使用者 alice，建立一個名為 `users_jaas.conf` 的檔案，內含以下內容。

```
KafkaClient {
   org.apache.kafka.common.security.scram.ScramLoginModule required
   username="alice"
   password="alice-secret";
};
```

4. 使用以下命令將 JAAS 組態檔案匯出為 `KAFKA_OPTS` 環境參數。

```
export KAFKA_OPTS=-Djava.security.auth.login.config=<path-to-jaas-file>/
users_jaas.conf
```

5. 在 `./tmp` 目錄中，建立名為 `kafka.client.truststore.jks` 的檔案。

6. 使用下列指令，將 JDK 金鑰存放區檔案從 JVM `cacerts` 資料夾複製到您在上一個步驟中建立的 `kafka.client.truststore.jks` 檔案。使用執行個體上 JDK 資料夾的名稱取代 *JDKFolder*。例如，您的 JDK 資料夾可能會命名為 `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. 在安裝 Apache Kafka 的 `bin` 目錄中，建立名為 `client_sasl.properties` 的用戶端屬性檔案，內含以下內容。此檔案會定義 SASL 機制和通訊協定。

```
security.protocol=SASL_SSL
sasl.mechanism=SCRAM-SHA-512
ssl.truststore.location=<path-to-keystore-file>/kafka.client.truststore.jks
```

8. 使用以下命令擷取引導代理程式字符串。以叢集*ClusterArn*的 Amazon 資源名稱 (ARN) 取代：

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

從命令的 JSON 結果中，儲存與名為 BootstrapBrokerStringSaslScram 的字串相關聯的值。

9. 若要產生資料到您建立的範例主題，請在用戶端機器上執行下列命令。將 *BootstrapBrokerStringSaslScram* 替換為您在上一步中檢索到的值。

```
<path-to-your-kafka-installation>/bin/kafka-console-producer.sh --broker-
list BootstrapBrokerStringSaslScram --topic ExampleTopicName --producer.config
 client_sasl.properties
```

10. 若要從您建立的主題取用資料，請在用戶端機器上執行下列命令。將 *BootstrapBrokerStringSaslScram* 替換為您之前獲得的值。

```
<path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
server BootstrapBrokerStringSaslScram --topic ExampleTopicName --from-beginning --
consumer.config client_sasl.properties
```

使用使用者

建立使用者：您可以在秘密中將使用者建立鍵值對。當您在 Secrets Manager 主控台中使用純文字選項時，您應該以下列格式指定登入憑證資料。

```
{
  "username": "alice",
  "password": "alice-secret"
}
```

撤銷使用者存取權：若要撤銷使用者存取叢集的憑證，建議您先移除或強制執行叢集上的 ACL，然後取消與秘密的關聯。其原因如下：

- 移除使用者並不會關閉現有的連線。

- 對秘密等變更最多需要 10 分鐘的時間傳播。

如需有關搭配 Amazon MSK 使用 ACL 的詳細資訊，請參閱 Apache Kafka ACL。

對於使用 ZooKeeper 模式的叢集，建議您限制 ZooKeeper 節點的存取權限，以防止使用者修改 ACL。如需詳細資訊，請參閱 控制對阿帕奇的存取 ZooKeeper。

限制

使用 SCRAM 秘密時，請注意以下限制：

- Amazon MSK 僅支援 SCRAM-SHA-512 身分驗證。
- 一個 Amazon MSK 叢集最多可以有 1000 個使用者。
- 你必須使用你 AWS KMS key 的秘密。您無法將採取預設 Secrets Manager 加密金鑰的秘密與 Amazon MSK 搭配使用。如需有關建立 KMS 金鑰的相關資訊，請參閱建立對稱加密 KMS 金鑰。
- 您無法透過 Secrets Manager 使用非對稱 KMS 金鑰。
- 使用此 BatchAssociateScramSecret作業一次最多可以關聯 10 個密碼與叢集。
- 與 Amazon MSK 叢集相關聯的秘密名稱必須使用字首 Amazon MSK_。
- 與 Amazon MSK 叢集相關聯的機密必須與叢集位於相同的 Amazon Web Services 帳戶和 AWS 區域。

Apache Kafka ACL

阿帕奇卡夫卡有一個可插拔的授權者，並附帶授權者實現。 out-of-box Amazon MSK 會在代理程式的 `server.properties` 檔案中啟用此授權。

阿帕奇卡夫卡 ACL 的格式為「主體 P 是 [允許/拒絕] 操作 O 從主機 H 在任何符合 RP 的資源 R 上」。 ResourcePattern 如果 RP 不匹配特定的資源 R，那麼 R 和 ACL 無關聯，因此除了超級使用者以外沒有其他人能夠存取 R。若要變更此 Apache Kafka 行為，請將屬性 `allow.everyone.if.no.acl.found` 設定為 "true"。Amazon MSK 預設設定為 "true"。這表示使用 Amazon MSK 叢集時，如果您未在資源上明確設定 ACL，則所有委託人都可以存取此資源。如果您啟用資源上的 ACL，只有授權的委託人可以存取它。如果您想要限制對主題的存取，並授權使用 TLS 相互驗證的用戶端，請使用 Apache Kafka 授權方 CLI 來新增 ACL。如需有關新增、移除和列出 ACL 的詳細資訊，請參閱 Kafka 授權命令列界面。

除了用戶端之外，您還需要授予所有代理程式存取您的主題，以便代理程式可以從主分區複製訊息。如果代理程式無法存取主題，主題的複寫就會失敗。

若要新增或移除主題的讀取和寫入權限

1. 將您的代理程式新增至 ACL 表格，以允許它們讀取所有具有 ACL 的主題。若要授予代理程式對主題的讀取權限，請在可與 MSK 叢集通訊的用戶端機器上執行下列命令。

   使用叢集的任何引導代理程式的 DNS 取代 *Distinguished-Name*，然後以星號 (*) 取代此辨別名稱中第一個句點之前的字串。例如，如果其中一個叢集的引導代理程式具有 DNS，b-6.mytestcluster.67281x.c4.kafka.us-east-1.amazonaws.com請使用 *.mytestcluster.67281x.c4.kafka.us-east-1.amazonaws.com 取代下列命令中的 *Distinguished-Name*。如需有關如何取得引導代理程式的資訊，請參閱the section called “取得引導代理程式”。

   ```
   <path-to-your-kafka-installation>/bin/kafka-acls.sh --authorizer-properties
    --bootstrap-server BootstrapServerString --add --allow-principal
    "User:CN=Distinguished-Name" --operation Read --group=* --topic Topic-Name
   ```

2. 若要授與主題的讀取權限，請在用戶端機器上執行下列命令。若您使用雙向 TLS 身分驗證，使用與建立私有金鑰時相同的 *Distinguished-Name*。

   ```
   <path-to-your-kafka-installation>/bin/kafka-acls.sh --authorizer-properties
    --bootstrap-server BootstrapServerString --add --allow-principal
    "User:CN=Distinguished-Name" --operation Read --group=* --topic Topic-Name
   ```

   若要移除讀取權限，您可以執行相同的命令，將 --add 取代為 --remove。

3. 若要授與主題的寫入權限，請在用戶端機器上執行下列命令。若您使用雙向 TLS 身分驗證，使用與建立私有金鑰時相同的 *Distinguished-Name*。

   ```
   <path-to-your-kafka-installation>/bin/kafka-acls.sh --authorizer-properties
    --bootstrap-server BootstrapServerString --add --allow-principal
    "User:CN=Distinguished-Name" --operation Write --topic Topic-Name
   ```

   若要移除寫入權限，您可以執行相同的命令，將 --add 取代為 --remove。

變更 Amazon MSK 叢集的安全群組

本頁說明如何變更現有 MSK 叢集的安全群組。您可能需要變更叢集的安全群組，才能提供對特定使用者集的存取權，或限制對叢集的存取。如需安全群組的詳細資訊，請參閱《Amazon VPC 使用者指南》中的 VPC 的安全群組。

1. 使用中的 ListNodesAPI 或列表節點命令 AWS CLI 來取得叢集中的代理程式清單。此操作的結果包括與代理程式相關聯的彈性網路介面 (ENI) ID。
2. 登入 AWS Management Console 並開啟 Amazon EC2 主控台，網址為 https://console.aws.amazon.com/ec2/。
3. 使用靠近螢幕右上角的下拉式清單，選取部署叢集的區域。
4. 在左側窗格中網路與安全下，選擇網路介面。
5. 選擇您在第一步中獲得的第一個 ENI。選擇畫面頂端的動作選單，然後選擇變更安全群組。將新的安全群組指派給此 ENI。針對您在第一個步驟中取得的每個 ENI 重複此步驟。

   > **Note**
   >
   > 您使用 Amazon EC2 主控台對叢集安全群組所做的變更，不會反映在網路設定下的 MSK 主控台中。

6. 設定新安全群組的規則，以確保您的用戶端可以存取代理程式。如需有關設定安全群組規則的詳細資訊，請參閱《Amazon VPC 使用者指南》中的新增、移除及更新規則。

> **Important**
>
> 如果您變更與叢集中代理程式相關聯的安全群組，然後將新代理程式新增至該叢集，Amazon MSK 會將新代理程式與叢集建立時相關聯的原始安全群組建立關聯。不過，若要讓叢集正常運作，叢集的所有代理程式都必須與相同的安全群組相關聯。因此，如果在變更安全群組後添加新代理程式，則必須再次執行之前的過程並更新新代理程式的 ENI。

控制對阿帕奇的存取 ZooKeeper

基於安全理由，您可以限制對屬於 Amazon MSK 叢集一部分之 Apache ZooKeeper 節點的存取。若要限制對這些節點的存取權，您可以將個別的安全群組指派給這些節點。然後，您可以決定誰可以存取該安全群組。

> **Important**
>
> 本節不適用於以 Kraft 模式執行的叢集。請參閱the section called “牛皮紙模式 ”。

本主題包含下列章節：

若要將 Apache ZooKeeper 節點放在單獨的安全性群組中

1. 取得叢集的 Apache ZooKeeper 連線字串。如要瞭解如何作業，請參閱the section called "ZooKeeper 模式"。連接字串包含 Apache ZooKeeper 節點的 DNS 名稱。
2. 使用 `host` 或 `ping` 之類的工具，來將您在上一個步驟中獲得的 DNS 名稱轉換為 IP 地址。儲存這些 IP 地址，因為稍後在此程序中需要這些資訊。
3. 登入 AWS Management Console 並開啟 Amazon EC2 主控台，網址為 https://console.aws.amazon.com/ec2/。
4. 在左側窗格中的 NETWORK & SECURITY (網路與安全) 下，選擇 Network Interfaces (網路界面)。
5. 在網路界面表格上方的搜尋欄位中，輸入叢集的名稱，然後輸入 return。這會將表格中顯示的網路界面數量限制為與叢集相關聯的那些界面。
6. 在清單中與第一個網路界面相對應的資料列開頭選取核取方塊。
7. 在頁面底部的詳細資訊窗格中，尋找 Primary private IPv4 IP (主要私有 IPv4 IP)。如果此 IP 位址符合您在此程序的第一個步驟中取得的其中一個 IP 位址，這表示此網路介面會指派給屬於叢集一部分的 Apache ZooKeeper 節點。否則，請取消選取此網路界面旁的核取方塊，然後選取清單中的下一個網路界面。網路界面的選取順序無關緊要。在接下來的步驟中，您將逐一對指派給 Apache ZooKeeper 節點的所有網路介面執行相同的作業。
8. 當您選取與 Apache ZooKeeper 節點對應的網路介面時，請選擇頁面頂端的「動作」功能表，然後選擇「變更安全性群組」。將新的安全群組指派至此網路界面。如需有關建立安全群組的詳細資訊，請參閱 Amazon VPC 文件中的建立安全群組。
9. 重複上一個步驟，將相同的新安全性群組指派給與叢集之 Apache ZooKeeper 節點相關聯的所有網路介面。
10. 您現在可以選擇可存取此全新安全群組的人員。如需有關設定安全群組規則的資訊，請參閱 Amazon VPC 文件中的新增、移除和更新規則。

搭配阿帕奇使用 TLS 安全性 ZooKeeper

您可以使用 TLS 安全性來在用戶端和 Apache ZooKeeper 節點之間傳輸過程中進行加密。若要使用 Apache ZooKeeper 節點實作 TLS 安全性，請執行下列動作：

- 叢集必須使用阿帕奇卡夫卡 2.5.1 版或更新版本，才能搭配 Apache 使用 TLS 安全性。 ZooKeeper
- 建立或設定叢集時啟用 TLS 安全功能。使用 Apache 卡夫卡 2.5.1 版或更新版本建立且啟用 TLS 的叢集會自動搭配 Apache 端點使用 TLS 安全性。 ZooKeeper 如需有關 TLS 安全功能設定的資訊，請參閱 如何開始使用加密？。
- 使用此DescribeCluster 作業擷取 TLS 阿帕奇 ZooKeeper 端點。
- 建立 Apache ZooKeeper 組態檔案，以便與kafka-configs.sh和kafka-acls.sh工具搭配使用，或搭配 ZooKeeper 殼層使用。對於每個工具，您都可以使用--zk-tls-config-file參數來指定 Apache ZooKeeper 配置。

  下面的例子顯示了一個典型的 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
  ```

- 對於其他命令 (例如kafka-topics)，您必須使用KAFKA_OPTS環境變數來設定 Apache ZooKeeper 參數。下列範例會示範如何設定KAFKA_OPTS環境變數，將 Apache ZooKeeper 參數傳遞至其他指令：

  ```
  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"
  ```

  設定 KAFKA_OPTS 環境變數之後，就可以正常使用 CLI 命令。下列範例會使用KAFKA_OPTS環境變數的 Apache ZooKeeper 組態，建立 Apache 卡夫卡主題：

  ```
  <path-to-your-kafka-installation>/bin/kafka-topics.sh --create --
  zookeeper ZooKeeperTLSConnectString --replication-factor 3 --partitions 1 --topic
   AWSKafkaTutorialTopic
  ```

> **Note**
>
> 您在 Apache ZooKeeper 配置文件中使用的參數名稱以及在KAFKA_OPTS環境變量中使用的參數名稱不一致。請注意您在組態檔案和 KAFKA_OPTS 環境變量中使用了哪些參數名稱。

如需有關使用 TLS 存取 Apache ZooKeeper 節點的詳細資訊，請參閱 KIP-515：讓 ZK 用戶端使用新的 TLS 支援的驗證。

日誌

您可以將 Apache Kafka 代理程式日誌傳遞到下列一或多個目的地類型：Amazon CloudWatch 日誌、Amazon S3、Amazon 資料 Firehose。您也可以使 AWS CloudTrail用記錄 Amazon MSK API 呼叫。

代理程式日誌

代理程式日誌可讓您針對 Apache Kafka 應用程式進行疑難排解，並分析與 MSK 叢集的通訊。您可以設定新的或現有的 MSK 叢集，將資訊層級代理程式記錄傳遞至下列一或多個目的地資源類型：日 CloudWatch 誌群組、S3 儲存貯體、Firehose 交付串流。然後，您可以透過 Firehose 將日誌資料從交付串流傳送至「 OpenSearch 服務」。您必須先建立目的地資源，才能設定叢集向其傳遞代理程式日誌。如果這些目的地資源尚未存在，Amazon MSK 不會為您建立它們。如需有關這三種目標資源類型以及如何建立這些資源的資訊，請參閱下列文件：

- Amazon CloudWatch 日誌
- Amazon Simple Storage Service (Amazon S3)
- Amazon 數據 Firehose

所需的許可

若要設定 Amazon MSK 代理程式日誌的目的地，您用於 Amazon MSK 動作的 IAM 身分必須具有 AWS 管理策略：亞馬遜 FullAccess 政策中所述的許可。

若要將代理程式日誌串流到 S3 儲存貯體，您也需要 `s3:PutBucketPolicy` 許可。如需有關 S3 儲存貯體政策的資訊，請參閱《Amazon S3 使用者指南》中的如何新增 S3 儲存貯體政策？ 如需有關 IAM 政策的一般資訊，請參閱《IAM 使用者指南》的存取管理。

使用 SSE-KMS 儲存貯體所需的 KMS 金鑰政策

如果您使用託管金鑰 (SSE-KMS) 和客戶 AWS KMS受管金鑰為 S3 儲存貯體啟用伺服器端加密，請將以下內容新增至 KMS 金鑰的金鑰政策，以便 Amazon MSK 可以將代理程式檔案寫入儲存貯體。

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

使用設定代理程式記錄檔 AWS Management Console

如果您要建立新叢集，請在監控區段中尋找代理程式日誌交付標題。您可以指定想要 Amazon MSK 向其傳遞代理程式日誌的目標。

針對現有叢集，請從您的叢集清單中選擇叢集，然後選擇屬性索引標籤。向下捲動到日誌交付區段，然後選擇其編輯按鈕。您可以指定想要 Amazon MSK 向其傳遞代理程式日誌的目標。

使用設定代理程式記錄檔 AWS CLI

當您使用 `create-cluster` 或 `update-monitoring` 命令時，可以選擇指定 `logging-info` 參數，並向其傳遞 JSON 結構，如以下範例所示。在此 JSON 中，所有三種目標類型都是選用的。

```
{
  "BrokerLogs": {
    "S3": {
      "Bucket": "ExampleBucketName",
      "Prefix": "ExamplePrefix",
      "Enabled": true
```

```
    },
    "Firehose": {
      "DeliveryStream": "ExampleDeliveryStreamName",
      "Enabled": true
    },
    "CloudWatchLogs": {
      "Enabled": true,
      "LogGroup": "ExampleLogGroupName"
    }
  }
}
```

使用 API 設定代理程式日誌

您可以在傳遞給CreateCluster或作業的 JSON 中指定選用`loggingInfo`結UpdateMonitoring構。

> **Note**
>
> 根據預設，啟用代理程式日誌後，Amazon MSK 會記錄 `INFO` 層級日誌至指定的目的地。然而，Apache Kafka 2.4.X 及更高版本的使用者可以將代理程式日誌層級動態設定為任何 log4j 日誌層級。如需有關動態設定代理程式日誌層級的相關資訊，請參閱 KIP-412: Extend Admin API to support dynamic application log levels。如果您將日誌級別動態設置為`DEBUG`或`TRACE`，我們建議您使用 Amazon S3 或 Firehose 作為日誌目的地。如果您使用 Lo CloudWatch gs 做為日誌目的地，並且動態啟用`DEBUG`或`TRACE`層級記錄，Amazon MSK 可能會持續提供日誌範例。這可能會大幅影響代理程式效能，只有在 `INFO` 日誌層級不夠詳細、無法判斷問題的根本原因時才應該使用方法。

使用 AWS CloudTrail記錄 API 呼叫

> **Note**
>
> AWS CloudTrail 只有在您使用時，日誌才可用IAM 存取控制於 Amazon MSK。

Amazon MSK 與服務整合在一起 AWS CloudTrail，該服務可提供 Amazon MSK 中使用者、角色或 AWS 服務所採取的動作記錄。 CloudTrail 擷取做為事件的 API 呼叫。擷取的呼叫包括從 Amazon MSK 主控台執行的呼叫，以及對 Amazon MSK API 操作發出的程式碼呼叫。它也會擷取 Apache Kafka 的動作，例如建立、變更主題和群組。

如果您建立追蹤，您可以啟用持續交付 CloudTrail 事件到 Amazon S3 儲存貯體，包括 Amazon MSK 的事件。如果您未設定追蹤，您仍然可以在 [事件歷程記錄] 中檢視 CloudTrail 主控台中最近的事件。使用收集的資訊 CloudTrail，您可以判斷向 Amazon MSK 或 Apache Kafka 動作提出的請求、提出請求的 IP 位址、提出請求的人員、提出請求的時間以及其他詳細資訊。

若要進一步了解 CloudTrail，包括如何設定和啟用它，請參閱AWS CloudTrail 使用者指南。

Amazon MSK 信息 CloudTrail

CloudTrail 當您創建帳戶時，您的 Amazon Web Services 帳戶已啟用。當受支援的事件活動發生在 MSK 叢集中時，該活動會與事件歷史記錄中的其他 AWS 服務 CloudTrail 事件一起記錄在事件中。您可以檢視、搜尋和下載 Amazon Web Services 帳戶中的最近事件。如需詳細資訊，請參閱檢視具有事 CloudTrail件記錄的事件。

如需 Amazon Web Services 帳戶中正在進行事件的記錄 (包含 Amazon MSK 的事件)，請建立追蹤。追蹤可 CloudTrail 將日誌檔交付到 Amazon S3 儲存貯體。根據預設，當您在主控台建立線索時，線索會套用到所有 區域。該追蹤會記錄來自 AWS 分割區中所有區域的事件，並將日誌檔案交付到您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 Amazon 服務，以進一步分析 CloudTrail 日誌中收集的事件資料並採取行動。如需詳細資訊，請參閱下列內容：

- 建立追蹤的概觀
- CloudTrail 支援的服務與整合
- 設定的 Amazon SNS 通知 CloudTrail
- 從多個區域接收 CloudTrail 記錄檔並從多個帳戶接收 CloudTrail 記錄檔

Amazon MSK 將所有 Amazon MSK 操作記錄為日 CloudTrail 誌檔中的事件。此外，它會記錄下列 Apache Kafka 動作。

- 卡夫卡集群：DescribeClusterDynamicConfiguration
- 卡夫卡集群：AlterClusterDynamicConfiguration
- 卡夫卡集群：CreateTopic
- 卡夫卡集群：DescribeTopicDynamicConfiguration
- 卡夫卡集群：AlterTopic
- 卡夫卡集群：AlterTopicDynamicConfiguration
- 卡夫卡集群：DeleteTopic

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 請求是以根使用者還是 AWS Identity and Access Management (IAM) 使用者登入資料提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 請求是否由其他 AWS 服務提出。

如需詳細資訊，請參閱CloudTrail 使 userIdentity 元素。

範例：Amazon MSK 日誌檔案項目

追蹤是一種組態，可讓事件以日誌檔的形式傳遞到您指定的 Amazon S3 儲存貯體。 CloudTrail 記錄檔包含一或多個記錄項目。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。 CloudTrail 日誌文件不是公共 API 調用和 Apache Kafka 操作的有序堆棧跟踪，因此它們不會以任何特定順序顯示。

下列範例顯示示範`DescribeCluster`和 `DeleteCluster` Amazon MSK 動作的 CloudTrail 記錄項目。

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

下列範例顯示示範kafka-cluster:CreateTopic動作的 CloudTrail 記錄項目。

```
{
```

```
  "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 的合規驗證

在 AWS 合規計劃中，第三方稽核人員會評估 Amazon Managed Streaming for Apache Kafka 的安全與合規情況。這些包含 PCI 和 HIPAA BAA。

如需特定合規計劃範圍內的 AWS 服務清單，請參閱合規計劃的 Amazon Web 服務範圍內的合。如需一般資訊，請參閱AWS 規範計劃AWS。

您可以使用下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載中的報告中的 AWS Artifact。

使用 Amazon MSK 時的合規責任取決於資料的敏感度、公司的合規目標以及適用的法律和法規。AWS 提供下列資源以協助遵循法規：

- 安全與合規快速入門指南：這些部署指南討論架構考量，並提供在 AWS上部署以安全及合規為重心之基準環境的步驟。
- 建構 HIPAA 安全性與合規性白皮書 — 本白皮書說明公司如何使用建立符合 HIPAA 標準的應用 AWS 程式。
- AWS 合規資源AWS — 此工作簿和指南集合可能適用於您的產業和所在地。
- 使用AWS Config 開發人員指南中的規則評估資源 — 此 AWS Config 服務會評估您的資源組態符合內部實務、產業準則和法規的程度。
- AWS Security Hub— 此 AWS 服務提供安全狀態的全面檢視，協助您檢查您 AWS 是否符合安全性產業標準和最佳做法。

Amazon Managed Streaming for Apache Kafka 的復原能力

AWS 全球基礎架構是圍繞區 AWS 域和可用區域建立的。 AWS 區域提供多個實體分離和隔離的可用區域，這些區域透過低延遲、高輸送量和高度備援的網路連線。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

如需區域和可用區域的相關 AWS 資訊，請參閱AWS 全域基礎結構。

Amazon Managed Streaming for Apache Kafka 的基礎設施安全性

作為受管服務，適用於 Apache Kafka 的 Amazon 受管串流受到 Amazon Web Services：安 AWS 全流程概觀白皮書中所述的全球網路安全程序的保護。

您可以使用 AWS 已發佈的 API 呼叫透過網路存取 Amazon MSK。用戶端必須支援 Transport Layer Security (TLS) 1.0 或更新版本。建議使用 TLS 1.2 或更新版本。用戶端也必須支援具備完美轉送私密 (PFS) 的密碼套件，例如臨時 Diffie-Hellman (DHE) 或橢圓曲線臨時 Diffie-Hellman (ECDHE)。現代系統 (如 Java 7 和更新版本) 大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 AWS Security Token Service (AWS STS) 來產生暫時安全憑證來簽署請求。

連線至 Amazon MSK 叢集

根據預設，用戶端只有位於與叢集相同的 VPC 時，才能存取 MSK 叢集。根據預設，Kafka 用戶端和 MSK 叢集之間的所有通訊都是私有的，並且串流資料永遠不會在網際網路周遊。如要從位於與叢集相同 VPC 的用戶端連線至 MSK 叢集，請確認叢集的安全群組具備接受來自用戶端安全群組流量的傳入規則。如需設定這些規則的資訊，請參閱安全群組規則。如需如何從位於與叢集相同 VPC 中 Amazon EC2 執行個體存取叢集的範例，請參閱開始使用。

若要從叢集 VPC 外部的用戶端連線至 MSK 叢集，請參閱從叢集的 VPC 內部 AWS 但外部存取。

主題

公用存取

Amazon MSK 可讓您選擇開啟對 MSK 叢集 (執行的是 Apache Kafka 2.6.0 或更新版本) 代理程式的公開存取。基於安全性考量，您無法在建立 MSK 叢集時開啟公開存取。但是，您可以更新現有叢集使其可公開存取。您也可以建立新叢集，然後對其進行更新，使其可公開存取。

您可以免費開啟 MSK 叢集的公用存取權，但是傳入和傳出叢集的 AWS 資料需支付標準資料傳輸費用。如需此定價的詳細資訊，請參閱 Amazon EC2 隨需定價。

若要開啟對叢集的公開存取，請先確保該叢集符合下列所有條件：

- 與叢集相關聯的子網路必須是公開的。這表示子網路必須具有相關聯的路由表，並連接網際網路閘道。如需有關建立和連接網際網路閘道的詳細資訊，請參閱《Amazon VPC 使用者指南》中的網際網路閘道。
- 未經身分驗證的存取控制必須關閉，且必須開啟下列至少其中一個存取控制方法：SASL/IAM、SASL/SCRAM、mTLS。如需有關如何更新叢集存取控制方法的詳細資訊，請參閱the section called “更新安全性”。
- 必須開啟叢集內的加密。建立叢集時，預設為「開啟」。對於建立時已關閉的叢集，無法開啟叢集內加密功能。因此，對於在叢集內加密關閉的情況下建立的叢集，無法開啟公開存取。
- 代理程式和用戶端之間的純文字流量必須關閉。如需有關如何在此項開啟時將其關閉的詳細資訊，請參閱the section called “更新安全性”。

- 如果您使用的是 SASL/SCRAM 或 mTLS 存取控制方法，必須為叢集設定 Apache Kafka ACL。為叢集設定 Apache Kafka ACL 之後，請更新叢集的組態，將叢集屬性 `allow.everyone.if.no.acl.found` 設為 false。如需有關更新叢集組態的資訊，請參閱the section called “組態操作”。如果您正在使用 IAM 存取控制，而且想要套用授權政策或更新授權政策，請參閱 the section called “IAM 存取控制”。如需有關 Apache Kafka ACL 的詳細資訊，請參閱 the section called “Apache Kafka ACL”。

確保 MSK 叢集符合上述條件後，您可以使用 AWS Management Console AWS CLI、或 Amazon MSK API 開啟公用存取。開啟對叢集的公開存取後，您可以為其取得公用引導代理程式字串。如需有關取得叢集引導代理程式的詳細資訊，請參閱the section called “取得引導代理程式”。

> **Important**
>
> 除了開啟公開存取之外，請確定叢集的安全群組具有傳入 TCP 規則，允許從 IP 地址進行公開存取。建議您盡可能嚴格設定這些規則。如需有關安全群組和傳入規則的詳細資訊，請參閱《Amazon VPC 使用者指南》中的 VPC 的安全群組。如需連接埠號碼，請參閱the section called “連接埠資訊”。如需有關變更叢集安全群組的說明，請參閱the section called “變更安全群組”。

> **Note**
>
> 如果您使用下列指示開啟公開存取，但仍無法存取該叢集，請參閱the section called “無法存取已開啟公開存取的叢集”。

使用主控台開啟公開存取

1. 登入 AWS Management Console，然後開啟 Amazon MSK 主控台，網址為 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。
2. 在叢集清單中，選擇您要開啟公開存取的叢集。
3. 選擇屬性索引標籤，然後尋找網路設定區段。
4. 選擇編輯公開存取。

使用開啟公用存取 AWS CLI

1. 執行下列 AWS CLI 命令，以 ARN *ClusterArn*和*#######*取代目前叢集版本。若要尋找叢集的目前版本，請使用DescribeCluster作業或描述 AWS CLI 叢集指令。範例版本為KTVPDKIKX0DER。

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

此 update-connectivity 命令的輸出如以下 JSON 範例所示。

```
{
    "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**
>
> 要關閉公共訪問，請使用類似的 AWS CLI 命令，但使用以下連接信息：
>
> ```
> '{"PublicAccess": {"Type": "DISABLED"}}'
> ```

2. 若要取得update-connectivity作業的結果，請執行下列命令，將 *ClusterOperationArn* 取代為您在命令輸出中取得的 update-connectivity ARN。

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

此 describe-cluster-operation 命令的輸出如以下 JSON 範例所示。

```
{
    "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` 具有值 `UPDATE_IN_PROGRESS`，請稍候一段時間，然後再次執行 `describe-cluster-operation` 命令。

使用 Amazon MSK API 開啟公開存取

- 若要使用 API 開啟或關閉叢集的公用存取權，請參閱UpdateConnectivity。

> **Note**
>
> 基於安全理由，Amazon MSK 不允許公開存取 Apache ZooKeeper 或 Kraft 控制器節點。

從叢集的 V AWS PC 內部但外部存取

若要從叢集的 Amazon VPC 內部 AWS 連線到 MSK 叢集，請使用以下選項。

Amazon VPC 對等互連

若要從位於與叢集 VPC 所在不同的 VPC 連線至 MSK 叢集，您可以建立兩個 VPC 之間的對等連線。如需 VPC 對等互連的資訊，請參閱 Amazon VPC 對等互連指南。

AWS Direct Connect

AWS Direct Connect AWS 透過標準的 1 Gigabit 或 10 Gigabit 乙太網路光纖纜線，將您的內部部署網路連結至。電纜的一端連接到路由器，另一端連接到 AWS Direct Connect 路由器。建立此連線後，您可以直接建立連至 AWS 雲端和 Amazon VPC 的虛擬界面，繞過網路路徑中的網際網路服務供應商。如需詳細資訊，請參閱 AWS Direct Connect。

AWS Transit Gateway

AWS Transit Gateway 是可讓您將 VPC 和內部部署網路連線到單一閘道的服務。如需如何使用 AWS Transit Gateway的資訊，請參閱 AWS Transit Gateway。

VPN 連線

您可以使用以下主題中所說明的 VPN 連線能力選項，將 MSK 叢集的 VPC 連線到遠端網路及使用者：VPN 連線。

REST 代理

您可以在於叢集 Amazon VPC 內執行的執行個體上安裝 REST 代理。REST 代理可讓生產者和取用者透過 HTTP API 請求與叢集通訊。

多區域多 VPC 連線

以下文件說明位於不同區域中多個 VPC 的連線選項：多區域多 VPC 連線。

單一區域多 VPC 私有連線

適用於 Apache Kafka (Amazon MSK AWS PrivateLink) 叢集的 Amazon 受管串流的多 VPC 私有連線 (由技術提供支援) 是一項功能，可讓您更快速地將託管在不同虛擬私有雲端 (VPC) 中的 Kafka 用戶端和帳戶連接到 Amazon MSK 叢集。 AWS

請參閱跨帳戶用戶端的單一區域多 VPC 連線。

EC2-經典網絡已退休

Amazon MSK 不再支援使用亞馬遜 EC2 傳統聯網執行的亞馬遜 EC2 執行個體。

請參閱 EC2-經典網絡正在退休 — 這裡是如何準備.

Amazon MSK 的單一區域多 VPC 私有連線

適用於 Apache Kafka (Amazon MSK AWS PrivateLink) 叢集的 Amazon 受管串流的多 VPC 私有連線 (由技術提供支援) 是一項功能，可讓您更快速地將託管在不同虛擬私有雲端 (VPC) 中的 Kafka 用戶端和帳戶連接到 Amazon MSK 叢集。 AWS

多 VPC 私有連線是一種受管解決方案，可簡化多 VPC 和跨帳戶連線的網路基礎設施。用戶端可以透過連線到 Amazon MSK 叢集， PrivateLink 同時保留 AWS 網路中的所有流量。適用於 Amazon MSK 叢集的多 VPC 私有連線功能，適用於 Amazon MSK 叢集的所有 AWS 區域皆可使用。

主題

什麼是多 VPC 私有連線？

Amazon MSK 的多 VPC 私有連線是一種連線選項，可讓您將託管在不同虛擬私有雲端 (VPC) 中的 Apache Kafka 用戶端和 AWS 帳戶連接到 MSK 叢集。

Amazon MSK 透過叢集政策簡化跨帳戶存取。這些原則可讓叢集擁有者授與其他 AWS 帳戶的權限，以建立 MSK 叢集的私人連線。

多 VPC 私有連線的優點

與其他連線解決方案相比，多 VPC 私有連線具備許多優點：

- 它可以自動化 AWS PrivateLink 連接解決方案的操作管理。
- 它允許在不同 VPC 連線之間重疊 IP，不需要像其他 VPC 連線解決方案那樣，要維護不重疊的 IP、複雜的對等互連以及路由表。

您可以針對 MSK 叢集使用叢集原則來定義哪些 AWS 帳戶具有設定 MSK 叢集的跨帳戶私人連線的權限。跨帳戶管理員可以將許可委派給適當的角色或使用者。與 IAM 用戶端身分驗證搭配使用時，您還可以使用叢集政策，為連線用戶端精細定義 Kafka 資料平面許可。

多 VPC 私有連線的需求和限制

請注意執行多 VPC 私有連線的 MSK 叢集需求：

- 僅 Apache Kafka 2.7.1 或更高版本支援多 VPC 私有連線。請確定與 MSK 叢集搭配使用的任何用戶端，執行的都是與該叢集相容的 Apache Kafka 版本。
- 多 VPC 私有連線支援 IAM、TLS 和 SASL/SCRAM 身分驗證類型。未經身分驗證的叢集無法使用多 VPC 私有連線。
- 如果您使用的是 SASL/SCRAM 或 mTLS 存取控制方法，必須為叢集設定 Apache Kafka ACL。首先，為叢集設定 Apache Kafka ACL。然後，更新叢集的組態，將叢集的屬性 `allow.everyone.if.no.acl.found` 設定為 false。如需有關更新叢集組態的資訊，請參閱the section called “組態操作”。如果您正在使用 IAM 存取控制，而且想要套用授權政策或更新授權政策，請參閱 the section called “IAM 存取控制”。如需有關 Apache Kafka ACL 的詳細資訊，請參閱 the section called “Apache Kafka ACL”。
- 多 VPC 私有連線不支援 t3.small 執行個體類型。
- 不支援跨 AWS 區域的多 VPC 私人連線，僅支援相同區域內的 AWS 帳戶。
- Amazon MSK 不支援與 Zookeeper 節點的多 VPC 私有連線。

開始使用多 VPC 私有連線

主題

- 步驟 1：在帳戶 A 中的 MSK 叢集上，針對叢集上的 IAM 身分驗證機制開啟多 VPC 連線
- 步驟 2：將叢集政策連接至 MSK 叢集
- 步驟 3：設定用戶端受管 VPC 連線的跨帳戶使用者動作

本教學課程使用常見使用案例做為範例，說明如何使用多重虛擬私人雲端連線，將 Apache Kafka 用戶端從叢集內部 AWS，但在 VPC 外部將 Apache Kafka 用戶端連線到 MSK 叢集。此程序需要跨帳戶

使用者為每個用戶端建立 MSK 受管的 VPC 連線和組態，包括必要的用戶端許可。此程序還要求 MSK 叢集擁有者啟用 MSK 叢集上的 PrivateLink 連線能力，並選取驗證配置以控制叢集的存取。

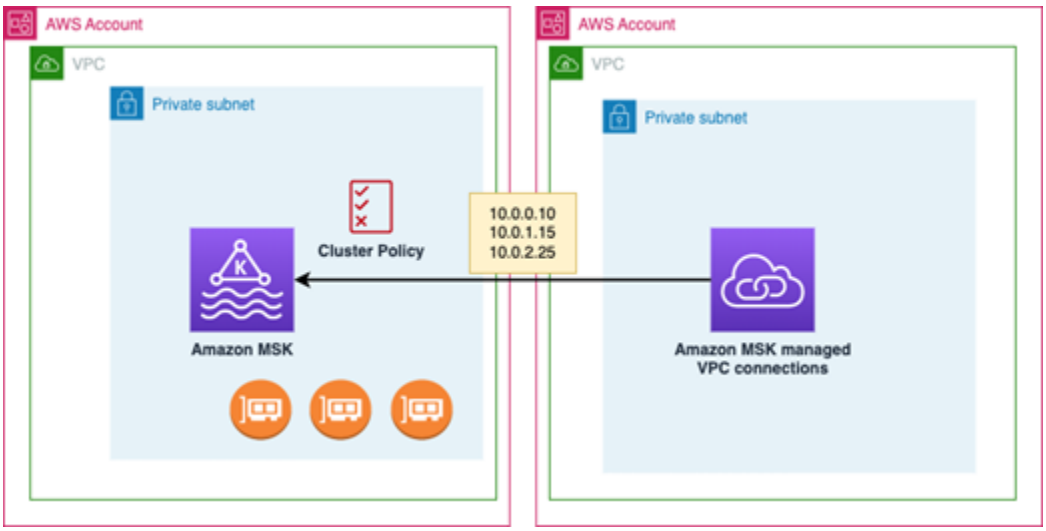
在本教程的其他部分，我們會選擇適用於此範例的選項。這並不表示只有這些選項才能用於設定 MSK 叢集或用戶端執行個體。

此使用案例的網路組態如下所示：

- 跨帳戶使用者 (Kafka 用戶端) 和 MSK 叢集位於相同的 AWS 網路/區域中，但在不同的帳戶中：
  - 帳戶 A 中的 MSK 叢集
  - 帳戶 B 中的 Kafka 用戶端
- 跨帳戶使用者將使用 IAM 身分驗證機制私有連線至 MSK 叢集。

本教學課程假設有使用 Apache Kafka 2.7.1 版或更高版本建立的 MSK 佈建叢集。MSK 叢集必須處於 ACTIVE 狀態，然後才能開始組態程序。為避免潛在的資料遺失或停機時間，將使用多 VPC 私有連線連接至叢集的用戶端，應該使用與此叢集相容的 Apache Kafka 版本。

下圖說明連接至不同帳戶中用戶端的 Amazon MSK 多 VPC 連線的架構。 AWS



步驟 1：在帳戶 A 中的 MSK 叢集上，針對叢集上的 IAM 身分驗證機制開啟多 VPC 連線

MSK 叢集擁有者需要在叢集建立並處於 ACTIVE 狀態之後，在 MSK 叢集上進行組態設定。

叢集擁有者針對將在叢集上處於作用中狀態的任何身分驗證機制，開啟 ACTIVE 叢集上的多 VPC 私有連線。這可以使用 UpdateSecurity API 或 MSK 控制台來完成。IAM、SASL/SCRAM 和 TLS 身分驗證機制支援多 VPC 私有連線。未經身分驗證的叢集無法啟用多 VPC 私有連線。

對於此使用案例，您將設定叢集以使用 IAM 身分驗證機制。

> **Note**
>
> 如果您將 MSK 叢集設定為使用 SASL/SCRAM 身分驗證機制，則必須使用 Apache Kafka ACL 屬性 "allow.everyone.if.no.acl.found=false"。請參閱 Apache Kafka ACL。

更新多 VPC 私有連線設定後，Amazon MSK 會以滾動式的方式，重新啟動更新代理程式組態的代理程式節點。這最多需要 30 分鐘或更久的時間才會完成。連線正在更新時，無法對叢集進行其他更新。

使用主控台為帳戶 A 中叢集上所選身分驗證機制開啟多 VPC

1. 在叢集所在的帳戶中，開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
2. 在導覽窗格的 MSK 叢集下，選擇叢集以顯示帳戶中的叢集清單。
3. 選取要設定多 VPC 私有連線的叢集。叢集必須處於 ACTIVE 狀態。
4. 選取叢集屬性索引標籤，然後移至網路設定。
5. 選取編輯下拉式選單，然後選取開啟多 VPC 連線。
6. 選取要為此叢集開啟的一個或多個身分驗證類型。對於此使用案例，請選取 IAM 角色型身分驗證。
7. 選取儲存變更。

Example -在叢集上開啟多 VPC 私有連線驗證配置的 UpdateConnectivity API

作為 MSK 主控台的替代方案，您可以使用 UpdateConnectivity API 開啟多 VPC 私人連線，並在 ACTIVE 叢集上設定驗證配置。下列範例顯示針對此叢集開啟 IAM 身分驗證機制。

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

```
  }
}
```

Amazon MSK 會建立私有連線所需的網路基礎設施。Amazon MSK 也會為需要私有連線的每個身分驗證類型，建立一組新的引導代理程式端點。請注意，純文字身分驗證機制不支援多 VPC 私有連線。

步驟 2：將叢集政策連接至 MSK 叢集

叢集擁有者可以將叢集政策 (也稱為資源型政策) 連接至 MSK 叢集，您將在其中開啟多 VPC 私有連線。叢集政策授予用戶端從另一個帳戶存取叢集的許可。在編輯叢集政策之前，您需要有權存取 MSK 叢集之帳戶的帳戶 ID。請參閱 Amazon MSK 如何與 IAM 搭配運作。

叢集擁有者必須將叢集政策連接至 MSK 叢集，該政策會授權帳戶 B 中的跨帳戶使用者取得該叢集的引導代理程式，並授權對帳戶 A 中 MSK 叢集的下列動作：

- CreateVpc連接
- GetBootstrap经纪人
- DescribeCluster
- DescribeClusterV2

Example

如需參考，以下是基本叢集政策的 JSON 範例，與在 MSK 主控台 IAM 政策編輯器中顯示的預設政策類似。

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

將叢集政策連接至 MSK 叢集

1. 在 Amazon MSK 主控台的 MSK 叢集下，選擇叢集。
2. 向下捲動至安全設定，然後選取編輯叢集政策。
3. 在主控台的編輯叢集政策畫面上，選取多 VPC 連線的基本政策。
4. 在帳戶 ID 欄位中，輸入每個有權存取此叢集之帳戶的帳戶 ID。輸入 ID 後，系統會自動將該 ID 複製到顯示的政策 JSON 語法中。在範例叢集政策中，帳戶 ID 為 123456789012。
5. 選取儲存變更。

如需有關叢集政策 API 的相關資訊，請參閱 Amazon MSK 資源型政策。

步驟 3：設定用戶端受管 VPC 連線的跨帳戶使用者動作

若要在與 MSK 叢集不同帳戶中的用戶端之間設定多 VPC 私有連線，則跨帳戶使用者需要為該用戶端建立受管 VPC 連線。重複此程序即可將多個用戶端連線至 MSK 叢集。就此使用案例而言，您只需設定一個用戶端。

用戶端可以使用受支援的身分驗證機制 IAM、SASL/SCRAM 或 TLS。每個受管 VPC 連線只能有一個相關聯的身分驗證機制。必須在用戶端將連線的 MSK 叢集上設定用戶端身分驗證機制。

在此使用案例中，請設定用戶端身分驗證機制，以便帳戶 B 中的用戶端使用 IAM 身分驗證機制。

必要條件

此程序需要下列項目：

- 先前建立的叢集政策，其會授予帳戶 B 中用戶端對帳戶 A 中的 MSK 叢集執行動作的許可。
- 附加至帳戶 B 中用戶端的身分識別原則，可授`ec2:CreateVPCEndpoint`與`kafka:CreateVpcConnection``ec2:CreateTags`、和`ec2:DescribeVpcAttribute`動作的權限。

Example

如需參考，以下是基本用戶端身分政策的 JSON 範例。

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

為帳戶 B 中的用戶端建立受管 VPC 連線

1. 從叢集系統管理員取得帳戶 A 中 MSK 叢集的叢集 ARN，以便帳戶 B 中的用戶端連線到該叢集。記下叢集 ARN 以供稍後使用。
2. 在帳戶 B 用戶端的 MSK 主控台中，選擇受管 VPC 連線，然後選擇建立連線。
3. 在連線設定窗格中，將叢集 ARN 貼至叢集 ARN 文字欄位，然後選擇驗證。
4. 為帳戶 B 中的用戶端選取身分驗證類型。在此使用案例中，請在建立用戶端 VPC 連線時選擇 IAM。
5. 選擇用戶端的 VPC。
6. 至少選擇兩個可用區域和關聯的子網路。您可以從 AWS 管理主控台叢集詳細資料或使用 DescribeClusterAPI 或描述叢集 CLI 命 AWS 令，取得可用區域識別碼。您為用戶端子網路指定的區域 ID 必須與叢集子網路的區域 ID 相符。如果遺失子網路的值，請先建立具有與 MSK 叢集相同區域 ID 的子網路。
7. 選擇此 VPC 連線的安全群組。您可以使用預設的安全群組。如需有關設定安全群組的詳細資訊，請參閱使用安全群組控制資源的流量。
8. 選取建立連線。
9. 若要從跨帳戶使用者的 MSK 主控台 (叢集詳細資訊 > 受管 VPC 連線) 取得新的引導代理程式字串清單，請參閱「叢集連線字串」下顯示的引導代理程式字串。在用戶端帳戶 B 中，您可以透過呼

叫 Bro GetBootstrapkers API 或在主控台叢集詳細資料中檢視啟動程式代理程式清單來檢視啟動程式代理程式清單。

10. 如下所示，更新與 VPC 連線關聯的安全群組：

    a. 設定 PrivateLink VPC 的輸入規則，以允許來自帳戶 B 網路之 IP 範圍的所有流量。

    b. [選用] 設定 MSK 叢集的傳出規則連線。在 VPC 主控台中選擇安全群組、編輯傳出規則，然後為連接埠範圍 14001-14100 新增自訂 TCP 流量規則。多 VPC 網路負載平衡器接聽 14001-14100 連接埠範圍。請參閱 Network Load Balancer。

11. 設定帳戶 B 中的用戶端，使用多 VPC 私有連線的新引導代理程式，以連線到帳戶 A 中的 MSK 叢集。請參閱生產和取用資料。

授權完成後，Amazon MSK 會為每個指定的 VPC 和身分驗證機制建立受管 VPC 連線。所選的安全群組會與每個連線相關聯。Amazon MSK 會設定此受管 VPC 連線，以私密連線至代理程式。您可以使用一組新的引導代理程式，私密連線到 Amazon MSK 叢集。

更新叢集上的授權機制

多 VPC 私有連線支援多種授權機制：SASL/SCRAM、IAM 和 TLS。叢集擁有者可以開啟/關閉一個或多個身分驗證機制的私有連線。此叢集必須處於 ACTIVE 狀態，才能執行此動作。

使用 Amazon MSK 主控台開啟身分驗證機制

1. 針對您要編輯的叢集，開啟位於 AWS Management Console 的 Amazon MSK 主控台。
2. 在導覽窗格的 MSK 叢集下，選擇叢集以顯示帳戶中的叢集清單。
3. 選取您要編輯的叢集。叢集必須處於 ACTIVE 狀態。
4. 選取叢集屬性索引標籤，然後移至網路設定。
5. 選取編輯下拉式選單，然後選取開啟多 VPC 連線，以開啟新的身分驗證機制。
6. 選取要為此叢集開啟的一個或多個身分驗證類型。
7. 選取開啟選取範圍。

開啟新的身分驗證機制時，也應為新的身分驗證機制建立新的受管 VPC 連線，並更新用戶端以使用新身分驗證機制特定的引導代理程式。

使用 Amazon MSK 主控台關閉身分驗證機制

> **Note**
>
> 關閉身分驗證機制的多 VPC 私有連線後，系統會刪除所有與連線相關的基礎設施，包括受管的 VPC 連線。

關閉身分驗證機制的多 VPC 私有連線後，用戶端上現有的 VPC 連線會變更為 INACTIVE 狀態，叢集端上的 Privatelink 基礎設施 (包括受管的 VPC 連線) 會遭到移除。跨帳戶使用者只能刪除非作用中的 VPC 連線。如果在該叢集上再次開啟私有連線，則跨帳戶使用者需要建立與該叢集的新連線。

1. 開啟位於 AWS Management Console 的 Amazon MSK 主控台。
2. 在導覽窗格的 MSK 叢集下，選擇叢集以顯示帳戶中的叢集清單。
3. 選取您要編輯的叢集。叢集必須處於 ACTIVE 狀態。
4. 選取叢集屬性索引標籤，然後移至網路設定。
5. 選取編輯下拉式選單，然後選取關閉多 VPC 連線 (以關閉身分驗證機制)。
6. 選取要為此叢集關閉的一個或多個身分驗證類型。
7. 選取關閉選取項目。

Example 使用 API 開啟/關閉身分驗證機制

作為 MSK 主控台的替代方案，您可以使用 UpdateConnectivity API 開啟多 VPC 私人連線，並在 ACTIVE 叢集上設定驗證配置。下列範例顯示開啟叢集的 SASL/SCRAM 和 IAM 身分驗證機制。

開啟新的身分驗證機制時，也應為新的身分驗證機制建立新的受管 VPC 連線，並更新用戶端以使用新身分驗證機制特定的引導代理程式。

關閉身分驗證機制的多 VPC 私有連線後，用戶端上現有的 VPC 連線會變更為 INACTIVE 狀態，叢集端上的 Privatelink 基礎設施 (包括受管的 VPC 連線) 會遭到移除。跨帳戶使用者只能刪除非作用中的 VPC 連線。如果在該叢集上再次開啟私有連線，則跨帳戶使用者需要建立與該叢集的新連線。

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

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

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

拒絕與 Amazon MSK 叢集的受管 VPC 連線

您可以從叢集管理員帳戶上的 Amazon MSK 主控台拒絕用戶端 VPC 連線。用戶端 VPC 連線必須處於 AVAILABLE 狀態，您才能拒絕此連線。您可能想要拒絕來自不再有權連線至叢集的用戶端的受管 VPC 連線。若要防止新的受管 VPC 連線連接至用戶端，請在叢集政策中拒絕對該用戶端的存取。在連線擁有者刪除連線前，遭拒的連線仍會產生成本。請參閱刪除與 Amazon MSK 叢集的受管 VPC 連線。

使用 MSK 主控台拒絕用戶端 VPC 連線

1. 開啟位於 AWS Management Console 的 Amazon MSK 主控台。
2. 在導覽窗格中，選取叢集並捲動至網路設定 > 用戶端 VPC 連線清單。
3. 選取您要拒絕的連線，然後選取拒絕用戶端 VPC 連線。
4. 確認您要拒絕所選的用戶端 VPC 連線。

若要使用 API 拒絕受管 VPC 連線，請使用 `RejectClientVpcConnection` API。

刪除與 Amazon MSK 叢集的受管 VPC 連線

跨帳戶使用者可以從用戶端帳戶主控台刪除 MSK 叢集的受管 VPC 連線。由於叢集擁有者使用者不是受管 VPC 連線的擁有者，因此無法從叢集管理員帳戶刪除該連線。VPC 連線刪除後就不會再產生成本。

使用 MSK 主控台刪除受管 VPC 連線

1. 在用戶端帳戶中，開啟位於 AWS Management Console 的 Amazon MSK 主控台。
2. 在導覽窗格中，選取受管 VPC 連線。
3. 從連線清單中選取您要刪除的連線。
4. 確認您要刪除該 VPC 連線。

若要使用 API 刪除受管 VPC 連線，請使用 `DeleteVpcConnection` API。

多 VPC 私有連線的許可

本節摘要說明使用多 VPC 私有連線功能的用戶端和叢集所需的許可。多 VPC 私有連線需要用戶端管理員在每個將與 MSK 叢集建立受管 VPC 連線的用戶端上建立許可。它還需要 MSK 叢集管理員啟用 MSK 叢集上的 PrivateLink 連線能力，並選取驗證配置來控制叢集的存取。

叢集身分驗證類型和主題存取許可

開啟 MSK 叢集啟用之身分驗證機制的多 VPC 私有連線功能。請參閱多 VPC 私有連線的需求和限制。如果您將 MSK 叢集設定為使用 SASL/SCRAM 身分驗證機制，則必須使用 Apache Kafka ACL 屬性 `allow.everyone.if.no.acl.found=false`。為叢集設定 Apache Kafka ACL 之後，請更新該叢集的組態，以將該叢集屬性 `allow.everyone.if.no.acl.found` 設為 false。如需有關更新叢集組態的資訊，請參閱Amazon MSK 組態操作。

跨帳戶叢集政策許可

如果 Kafka 用戶端所在的 AWS 帳戶與 MSK 叢集不同，請將叢集型原則附加至授權用戶端 root 使用者進行跨帳戶連線的 MSK 叢集。您可以使用 MSK 主控台中的 IAM 政策編輯器 (叢集安全設定 > 編輯叢集政策)，編輯多 VPC 叢集政策，或使用下列 API 來管理叢集政策：

PutCluster政策

將叢集政策連接至叢集。您可以使用此 API，來建立或更新指定的 MSK 叢集政策。如果您要更新政策，則必須在請求承載中包含 currentVersion 欄位。

GetCluster政策

擷取與叢集連接之叢集政策文件的 JSON 文字。

DeleteCluster政策

刪除叢集政策。

以下是基本叢集政策的 JSON 範例，其與在 MSK 主控台 IAM 政策編輯器中顯示的政策類似。

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

用戶端對與 MSK 叢集之多 VPC 私有連線的許可

若要在 Kafka 用戶端和 MSK 叢集之間設定多 VPC 私有連線，用戶端需要連接的身分政策，以授予在用戶端上進行 `kafka:CreateVpcConnection`、`ec2:CreateTags` 和 `ec2:CreateVPCEndpoint` 動作的許可。如需參考，以下是基本用戶端身分政策的 JSON 範例。

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

```
      "Effect": "Allow",
      "Action": [
        "kafka:CreateVpcConnection",
        "ec2:CreateTags",
        "ec2:CreateVPCEndpoint"
      ],
      "Resource": "*"
    }
  ]
}
```

連接埠資訊

使用下列連接埠號碼，讓 Amazon MSK 能夠與用戶端機器通訊：

- 若要使用純文字與代理程式通訊，請使用連接埠 9092。
- 要與具有 TLS 加密的代理程序進行通信，請使用端口 9094 從內部進行訪問， AWS 並使用端口 9194 進行公共訪問。
- 要與具有 SASL/SCRAM 的經紀人進行通信，請使用端口 9096 從內部訪問， AWS 並使用端口 9196 進行公共訪問。
- 若要與已設定使用的叢集中的代理程式通訊the section called “IAM 存取控制”，請使用連接埠 9098 從內 AWS 部存取，使用連接埠 9198 進行公用存取。
- 若要使用 TLS 加密與阿帕奇 ZooKeeper 通訊，請使用連接埠 2182。默認情況下，阿帕奇 ZooKeeper 節點使用端口 2181。

遷移至 Amazon MSK 叢集

Amazon MSK Replicator 可用於 MSK 叢集遷移。請參閱什麼是 Amazon MSK Replicator？。或者，您也可以使用 Apache MirrorMaker 2.0 從非 MSK 叢集遷移到 Amazon MSK 叢集。有關如何執行此操作的範例，請參閱使用將現場部署 Apache 卡夫卡叢集遷移到 Amazon MSK。 MirrorMaker如需如何使用的詳細資訊 MirrorMaker，請參閱 Apache Kafka 說明文件中的叢集之間鏡像資料。我們建議 MirrorMaker 在高可用性組態中進行設定。

使用移轉至 MSK 叢集時應 MirrorMaker 遵循的步驟大綱

1. 建立目的地 MSK 叢集
2. MirrorMaker 從與目標叢集相同的 Amazon VPC 內的 Amazon EC2 執行個體開始。
3. 檢查 MirrorMaker 滯後。
4. MirrorMaker 趕上之後，使用 MSK 叢集啟動程式代理程式將生產者和消費者重新導向至新叢集。
5. 關閉 MirrorMaker。

將您的 Apache Kafka 叢集遷移到 Amazon MSK

假設你有一個名為 CLUSTER_ONPREM 的 Apache Kafka 叢集。該叢集會填入主題和資料。如果您想要將該叢集遷移到新建立且名為 CLUSTER_AWSMSK 的 Amazon MSK 叢集，此程序會提供您必須遵循之步驟的高階檢視。

將現有的 Apache Kafka 叢集遷移到 Amazon MSK

1. 在中 CLUSTER_AWSMSK，建立您要遷移的所有主題。

   您無法用 MirrorMaker 於此步驟，因為它不會自動以正確的複寫層級重新建立您要移轉的主題。您可以在 Amazon MSK 中建立相同的主題，並具有與 CLUSTER_ONPREM 中相同的複寫係數和分區數量。您也可以建立具有不同複寫因素和分割區數目的主題。
2. MirrorMaker 從具有讀取權限CLUSTER_ONPREM和寫入權限的執行個體開始CLUSTER_AWSMSK。
3. 執行下列命令以鏡像所有主題：

```
<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 指定 CLUSTER_ONPREM 中的引導代理程式；例如，bootstrap.servers=localhost:9092。並config/mirrormaker-producer.properties指向 CLUSTER_ 中的引導程序代理程序AWSMSK; 例如，. bootstrap.servers=10.0.0.237:9092,10.0.2.196:9092,10.0.1.233:9092

4. 繼續在後台 MirrorMaker 運行，並繼續使用CLUSTER_ONPREM。 MirrorMaker 鏡像所有新資料。
5. 檢查鏡像的進度，方法 MirrorMaker 是檢查每個主題的最後一個偏移與消耗的目前偏移之間的延遲。

   請記住，只 MirrorMaker 是使用消費者和生產者。所以，你可以使用 kafka-consumer-groups.sh 工具來檢查延遲。若要尋找取用者群組名稱，請在 mirrormaker-consumer.properties 檔案內尋找 group.id，然後使用該值。如果檔案中沒有這樣的金鑰，則你可以建立。例如，設定 group.id=mirrormaker-consumer-group。
6. MirrorMaker 完成鏡像所有主題後，停止所有生產者和消費者，然後停止 MirrorMaker。然後通過變更生產者和取用者引導代理程式的值，重定導向到 CLUSTER_AWSMSK 叢集。重新啟動 CLUSTER_AWSMSK 上的所有生產者和取用者。

從一個 Amazon MSK 叢集遷移至另一個叢集

您可以使用 Apache MirrorMaker 2.0 從非 MSK 叢集移轉至 MSK 叢集。例如，您可以從 Apache Kafka 的一個版本遷移到另一個版本。有關如何執行此操作的範例，請參閱使用將現場部署 Apache 卡夫卡叢集遷移到 Amazon MSK。 MirrorMakerAmazon MSK Replicator 也可用於 MSK 叢集遷移。如需有關 Amazon MSK Replicator 的詳細資訊，請參閱 MSK Replicator。

MirrorMaker 1.0 最佳做法

此最佳做法清單適用於 MirrorMaker 1.0。

- 在目 MirrorMaker 的地叢集上執行。如此一來，如果發生網路問題，訊息仍可在來源叢集中使用。如果您在來源叢集 MirrorMaker 上執行，且事件會在生產者中緩衝，而且發生網路問題，事件可能會遺失。
- 如果傳輸過程中需要加密，請在來源叢集中執行。
- 針對取用者，設定 auto.commit.enabled=false
- 針對生產者，設定
  - max.in.flight.requests.per.connection=1

  - retries = 擷取最大值
  - acks = 全部
  - max.block.ms = Long.Max_Value
- 針對高生產者輸送量：
  - 緩衝訊息和填充訊息批次 - 調校 buffer.memory、batch.size、linger.ms
  - 調校插槽緩衝區 - receive.buffer.bytes、send.buffer.bytes
- 為了避免數據丟失，請關閉源代碼的 auto 提交， MirrorMaker 以便可以控制提交，這通常在從目標集群接收到 ack 之後執行。如果生產者具有 acks=all，且目的地叢集的 min.insync.replicas 設定為 1 以上，則在取用者在來源確認偏移量之前，訊息會保留在目的地的多個代理程式上。 MirrorMaker
- 如果順序很重要，您可以將 retries 設定為 0。或者，對於生產環境，將最大傳輸中連線設定為 1，以確保在批次失敗時發出的批次會按順序遞交。如此一來，每個發送的批次都會重試，直到下一個批次發送出去。如果 max.block.ms 未設定為最大值，且生產者緩衝區已滿，則可能會有資料遺失 (取決於某些其他設定)。這可以阻止取用者和返回壓力。
- 針對高輸送量
  - 增加緩衝區。記憶體。
  - 增加批次大小。
  - 調校 linger.ms 以允許批次填充。這也允許較佳的壓縮、較少的網路頻寬使用量，以及較少的叢集儲存量。這會導致增加保留。
  - 監控 CPU 和記憶體用量。
- 針對高取用者輸送量
  - 增加每 MirrorMaker 個進程的線程/消費者數量-num.stream。
  - 在增加執行緒以實現高可用性之前，先增加機器之間的 MirrorMaker 處理序數量。
  - 首先在同一台機器上增加 MirrorMaker 進程數，然後在不同的計算機上（具有相同的組 ID）上增加進程的數量。
  - 隔離具有非常高輸送量的主題，並使用不同的 MirrorMaker 執行個體。
- 針對管理和設定
  - 使用 AWS CloudFormation 和配置管理工具，如廚師和 Ansible。
  - 使用 Amazon EFS 掛載來維持所有 Amazon EC2 執行個體可存取的所有組態檔案。
  - 使用容器輕鬆擴展和管理 MirrorMaker執行個體。
- 通常情況下，需要多個消費者才能使生產者飽和。 MirrorMaker所以，設定多個取用者。首先，將它們設定在不同的機器上以提供高可用性。然後，將個別機器擴展為每個分割區具有取用者，讓取用者平均分配在各個機器之間。

- 對於高輸送量的擷取和傳遞，請調校接收和傳送緩衝區，因為它們的預設值可能太低。若要取得最大效能，請確定串流總數 (num.stream) 符合嘗試複製到目的地叢集的 MirrorMaker 所有主題分割區。

MirrorMaker 2.* 優勢

- 使用 Apache Kafka Connect 框架與生態系統。
- 偵測新主題和分割區。
- 自動在叢集間同步主題組態。
- 支援「作用中/作用中」叢集對，以及任何數量的作用中叢集。
- 提供新的指標，包括跨多個資料中心和叢集的 end-to-end 複寫延遲。
- 發出在叢集間遷移取用者所需要的位移，並提供位移平移的工具。
- 與每 MirrorMaker 個 1.* 程序的低階產生器/用戶特性相比，支援高階組態檔案，可在單一位置指定多個叢集和複寫流程。

監控 Amazon MSK 叢集

Amazon MSK 可透過數種方式協助您監控 Amazon MSK 叢集的狀態。

- Amazon MSK 可在叢集即將達到儲存容量限制時自動傳送儲存容量警示，協助您監控磁碟儲存容量。這些警示也會針對偵測到的問題提供最佳步驟建議。這可協助您識別並快速解決磁碟容量問題，以免其近一步嚴重化。Amazon MSK 會自動將這些警示傳送至您帳戶的 Amazon MSK 主控台 AWS Health Dashboard EventBridge、Amazon 和電子郵件聯絡 AWS 人。如需儲存容量警示的資訊，請參閱 Amazon MSK 儲存容量警示。
- Amazon MSK 收集 Apache Kafka 指標，並將其發送到 Amazon，您可以 CloudWatch 在其中查看它們。如需有關 Apache Kafka 指標的詳細資訊 (包括 Amazon MSK 呈現的指標)，請參閱 Apache Kafka 文件中的 Monitoring。
- 您也可以使用開放原始碼監控應用程式 Prometheus 監控您的 MSK 叢集。如需 Prometheus 的資訊，請參閱 Prometheus 文件中的 Overview。如要了解如何使用 Prometheus 監控您的叢集，請參閱 the section called “使用 Prometheus 進行開放式監控”。

主題

用於監控的 Amazon MSK 指標 CloudWatch

Amazon MSK 與 Amazon 整合， CloudWatch 因此您可以收集、檢視和分析 Amazon MSK 叢集的 CloudWatch指標。您為 MSK 叢集設定的指標會自動收集並推送至 CloudWatch。您可以將 MSK 叢集的監控層級設定為下列之一：`DEFAULT`、`PER_BROKER`、`PER_TOPIC_PER_BROKER`、`PER_TOPIC_PER_PARTITION`。下列章節中的表格顯示從每個監控層次開始可用的所有指標。

> **Note**
>
> 3.6.0 及更高版本中，某些用於 CloudWatch 監控的 Amazon MSK 指標的名稱已變更。使用新名稱來監控這些指標。針對已變更名稱的指標，下表顯示 3.6.0 版本及更高版本中使用的名稱，後方則顯示 2.8.2.tiered 版本中的名稱。

DEFAULT 層級指標是免費的。其他指標的定價請參閱 Amazon 定 CloudWatch價頁面。

DEFAULT 層級監控

下表所述的指標可在 DEFAULT 監控層級取得。他們是免費的。

DEFAULT 監控層級提供的指標

名稱	可見時	維度	描述
ActiveController Count	叢集到達 ACTIVE 狀態之後。	叢集名稱	每個叢集在任何時間應只能有一個控制器，處於作用中狀態。
BurstBalance	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	叢集中 EBS 磁碟區輸入輸出高載額度的餘額。用來調查延遲或輸送量降低的情況。 當磁碟區基準效能高於最大高載效能時，系統不會報告 EBS 磁碟區的 BurstBalance 。如需詳細資訊，請參閱 I/O 額度和高載效能。
BytesInPerSec	建立主題之後。	叢集名稱、代理程式 ID、主題	從用戶端接收的每秒位元組數量。此指標可用於每個代理程式和每個主題。
BytesOutPerSec	建立主題之後。	叢集名稱、代理程式	傳送至用戶端的每秒位元組數量。此指標可用於每個代理程式和每個主題。

名稱	可見時	維度	描述
		ID、主題	
ClientConnectionCount	叢集到達 ACTIVE 狀態之後。	叢集名稱、代理程式 ID、用戶端身分驗證	主動進行身分驗證的用戶端連線數。
ConnectionCount	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	主動進行身分驗證、未進行身分驗證和代理程式間的連線數。
CPUCreditBalance	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	自代理程式啟動後，累積獲得的 CPU 點數數量。獲得額度後，額度會在額度餘額中累積，並在支付額度時，從額度餘額中移出。如果您耗盡了 CPU 點數餘額，其可能會對叢集效能產生負面影響。您可以採取措施來減少 CPU 負載。例如，您可以減少用戶端請求數目，或將代理程式類型更新為 M5。
CpuIdle	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	CPU 閒置時間的百分比。
CpuIoWait	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	擱置磁碟操作期間 CPU 閒置時間的百分比。

名稱	可見時	維度	描述
CpuSystem	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	核心空間中的 CPU 百分比。
CpuUser	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	使用者空間中的 CPU 百分比。
GlobalPartitionCount	叢集到達 ACTIVE 狀態之後。	叢集名稱	叢集中所有主題 (不包括複本) 的分區數目。由於GlobalPartitionCount 不包含複本，因此PartitionCount 值的總和可能 GlobalPartitionCount 會高於主題的複寫因子大於 1 時。
GlobalTopicCount	叢集到達 ACTIVE 狀態之後。	叢集名稱	叢集中所有代理程式的主題總數。
EstimatedMaxTimeLag	取用者群組取用一個主題之後。	取用者群組、主題	估計耗盡 MaxOffsetLag 的時間 (秒)。
KafkaAppLogsDiskUsed	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	用於應用程式記錄檔的磁碟空間百分比。
KafkaDataLogsDiskUsed (Cluster Name, Broker ID 維度)	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	用於資料日誌的磁碟空間百分比。

名稱	可見時	維度	描述
KafkaData LogsDiskUsed (Cluster Name 維度)	叢集到達 ACTIVE 狀態之後。	叢集名稱	用於資料日誌的磁碟空間百分比。
LeaderCount	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	每個代理程式的分區領導者總數 (不包括複本)。
MaxOffsetLag	取用者群組取用一個主題之後。	取用者群組、主題	主題中所有分區的最大偏移延遲。
MemoryBuffered	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	代理程式的緩衝記憶體大小 (以位元組為單位)。
MemoryCached	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	代理程式的快取記憶體大小 (以位元組為單位)。
MemoryFree	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	可用且可供代理程式使用的記憶體大小 (以位元組為單位)。
HeapMemoryAfterGC	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	垃圾回收之後使用中的總堆積記憶體百分比。

名稱	可見時	維度	描述
MemoryUsed	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	代理程式使用的記憶體大小 (以位元組為單位)。
MessagesInPerSec	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	代理程式每秒內送訊息的數量。
NetworkRxDropped	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	已捨棄接收套件的數目。
NetworkRxErrors	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	網路接收代理程式的錯誤數目。
NetworkRxPackets	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	代理程式接收的封包數量。
NetworkTxDropped	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	已捨棄的傳輸套件數目。
NetworkTxErrors	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	代理程式的網路傳輸錯誤數目。

名稱	可見時	維度	描述
NetworkTx Packets	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	代理程式傳輸的封包數目。
OfflinePa rtitionsCount	叢集到達 ACTIVE 狀態之後。	叢集名稱	叢集中離線的磁碟分割區總數。
PartitionCount	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	每個代理程式的主題分區總數 (包括複本)。
ProduceTo talTimeMsMean	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	平均產生的時間 (以毫秒為單位)。
RequestBy tesMean	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	代理程式請求位元組的平均數。
RequestTime	套用請求調節之後。	叢集名稱，代理程式 ID	在代理程式網路和 I/O 執行緒間處理請求所花費的平均時間 (毫秒)。
RootDiskUsed	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	代理程式所使用的根磁碟百分比。
SumOffsetLag	取用者群組取用一個主題之後。	取用者群組、主題	主題中所有分區的彙整偏移延遲。

名稱	可見時	維度	描述
SwapFree	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	代理程式可用的交換記憶體大小 (以位元組為單位)。
SwapUsed	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	代理程式使用中交換記憶體大小 (以位元組為單位)。
TrafficShaping	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	高層級指標，指出因超過網路配置而成形 (丟棄或排入佇列) 的封包數。PER_BROKER 指標可提供更精細的資訊。
UnderMinIsrPartitionCount	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	代理程式在 minIsr 分割區下的數量。
UnderReplicatedPartitions	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	代理程式複製不足的分割區數量。
ZooKeeperRequestLatencyMsMean	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	對於 ZooKeeper基於叢集。來自代理程式之 Apache ZooKeeper 要求的平均延遲 (以毫秒為單位
ZooKeeperSessionState	叢集到達 ACTIVE 狀態之後。	叢集名稱，代理程式 ID	對於 ZooKeeper基於叢集。代理人的 ZooKeeper 工作階段的連線狀態，可能是下列其中一項：無 _ 已連線：'0.0'，關聯：'0.1'，連線：'0.5'，連線只讀：'0.8'，已連線：'1.0'，關閉：'5.0'，AUTH_失敗：'10.0'。

PER_BROKER 層級監控

將監控層級設定為時 PER_BROKER，除了所有 DEFAULT 層級指標以外，還會取得下列表格描述的指標。您為下列表格中的指標付費，而 DEFAULT 層級指標仍然是免費的。此表格中的指標包含下列維度：Cluster Name (叢集名稱)、Broker ID (代理程式 ID)。

從 **PER_BROKER** 監控層級開始可額外使用的指標

名稱	可見時	描述
BwInAllowanceExceeded	叢集到達 ACTIVE 狀態之後。	因傳入的彙總頻寬超過代理程式的上限而成形的封包數。
BwOutAllowanceExceeded	叢集到達 ACTIVE 狀態之後。	因傳出的彙總頻寬超過代理程式的上限而成形的封包數。
ConnTrackAllowanceExceeded	叢集到達 ACTIVE 狀態之後。	因連線追蹤超過代理程式的上限而成形的封包數。連線追蹤與安全群組相關，會追蹤每個建立的連線，以確保傳回封包如預期般交付。
ConnectionCloseRate	叢集到達 ACTIVE 狀態之後。	每個接聽程式每秒關閉的連線數。此數字會針對每個接聽程式彙總，並針對用戶端接聽程式進行篩選。
ConnectionCreationRate	叢集到達 ACTIVE 狀態之後。	每一個接聽程式每秒建立的新連線數。此數字會針對每個接聽程式彙總，並針對用戶端接聽程式進行篩選。
CpuCreditUsage	叢集到達 ACTIVE 狀態之後。	由代理程式使用的 CPU 點數。如果您耗盡了 CPU 點數餘額，其可能會對叢集效能產生負面影響。您可以採取措施來減少 CPU 負載。例如，您可以減少用戶端請求數目，或將代理程式類型更新為 M5。
FetchConsumerLocalTimeMsMean	有一個生產者/取用者之後。	領導者處理取用者請求的平均時間 (毫秒)。

名稱	可見時	描述
FetchConsumerRequestQueueTimeMsMean	有一個生產者/取用者之後。	取用者請求在佇列中等待的平均時間 (毫秒)。
FetchConsumerResponseQueueTimeMsMean	有一個生產者/取用者之後。	取用者請求在回應佇列中等待的平均時間 (毫秒)。
FetchConsumerResponseSendTimeMsMean	有一個生產者/取用者之後。	取用者傳送回應的平均時間 (毫秒)。
FetchConsumerTotalTimeMsMean	有一個生產者/取用者之後。	取用者從代理程式擷取資料時花費的平均總時間 (毫秒)。
FetchFollowerLocalTimeMsMean	有一個生產者/取用者之後。	領導者處理追隨者請求的平均時間 (以毫秒為單位)。
FetchFollowerRequestQueueTimeMsMean	有一個生產者/取用者之後。	追隨者請求在請求佇列中等待的平均時間 (以毫秒為單位)。
FetchFollowerResponseQueueTimeMsMean	有一個生產者/取用者之後。	追隨者請求在回應佇列中等待的平均時間 (以毫秒為單位)。
FetchFollowerResponseSendTimeMsMean	有一個生產者/取用者之後。	追隨者傳送回應的平均時間 (以毫秒為單位)。
FetchFollowerTotalTimeMsMean	有一個生產者/取用者之後。	追隨者花費在從代理程式獲取數據的平均總時間 (以毫秒為單位)。
FetchMessageConversionsPerSec	建立主題之後。	代理程式擷取訊息轉換的次數 (以秒為單位)。
FetchThrottleByteRate	套用頻寬調節之後。	每秒調節的位元組數量。
FetchThrottleQueueSize	套用頻寬調節之後。	調節佇列中的訊息數量。
FetchThrottleTime	套用頻寬調節之後。	平均擷取調節時間 (以毫秒為單位)。

名稱	可見時	描述
IAMNumberOfConnectionRequests	叢集到達 ACTIVE 狀態之後。	每秒 IAM 身份驗證請求的數量。
IAMTooManyConnections	叢集到達 ACTIVE 狀態之後。	嘗試超過 100 的連線數。0 表示連線數目在限制範圍內。如果 >0，則超過油門限制，您需要減少連接數量。
NetworkProcessorAvgIdlePercent	叢集到達 ACTIVE 狀態之後。	網路處理器閒置時間的平均百分比。
PpsAllowanceExceeded	叢集到達 ACTIVE 狀態之後。	因雙向 PPS 超過代理程式的上限而成形的封包數。
ProduceLocalTimeMsMean	叢集到達 ACTIVE 狀態之後。	領導者處理請求的平均時間 (毫秒)。
ProduceMessageConversionsPerSec	建立主題之後。	代理程式產生訊息轉換的次數 (以秒為單位)。
ProduceMessageConversionsTimeMsMean	叢集到達 ACTIVE 狀態之後。	訊息格式轉換所花費的平均時間 (以毫秒為單位)。
ProduceRequestQueueTimeMsMean	叢集到達 ACTIVE 狀態之後。	請求訊息在佇列中花費的平均時間 (以毫秒為單位)。
ProduceResponseQueueTimeMsMean	叢集到達 ACTIVE 狀態之後。	回應訊息在佇列中花費的平均時間 (以毫秒為單位)。
ProduceResponseSendTimeMsMean	叢集到達 ACTIVE 狀態之後。	傳送回應訊息所花費的平均時間 (以毫秒為單位)。
ProduceThrottleByteRate	套用頻寬調節之後。	每秒調節的位元組數量。
ProduceThrottleQueueSize	套用頻寬調節之後。	調節佇列中的訊息數量。
ProduceThrottleTime	套用頻寬調節之後。	平均產生調節時間 (以毫秒為單位)。

名稱	可見時	描述
ProduceTotalTimeMsMean	叢集到達 ACTIVE 狀態之後。	平均產生的時間 (以毫秒為單位)。
RemoteFetchBytesPerSec (RemoteBytesInPerSec in v2.8.2.tiered)	有一個生產者/取用者之後。	為回應取用者擷取而從分層儲存傳輸的位元組總數。此指標包括會產生下游資料傳輸流量的所有主題分區。類別：流量和錯誤率。這是一個 KIP-405 指標。
RemoteCopyBytesPerSec (RemoteBytesOutPerSec in v2.8.2.tiered)	有一個生產者/取用者之後。	傳輸至分層儲存的位元組總數，包括來自日誌區段、索引和其他輔助檔案的資料。此指標包含會產生上游資料傳輸流量的所有主題分區。類別：流量和錯誤率。這是一個 KIP-405 指標。
RemoteLogManagerTasksAvgIdlePercent	叢集到達 ACTIVE 狀態之後。	遠端日誌管理器閒置的平均時間百分比。遠端日誌管理器會將資料從代理程式傳輸到分層儲存。類別：內部活動。這是一個 KIP-405 指標。
RemoteLogReaderAvgIdlePercent	叢集到達 ACTIVE 狀態之後。	遠端日誌讀取器閒置的平均時間百分比。遠端日誌讀取器會將資料從遠端儲存傳輸到代理程式，以回應取用者擷取。類別：內部活動。這是一個 KIP-405 指標。
RemoteLogReaderTaskQueueSize	叢集到達 ACTIVE 狀態之後。	正在等待排程的負責從分層儲存進行讀取的任務數。類別：內部活動。這是一個 KIP-405 指標。
RemoteFetchErrorsPerSec (RemoteReadErrorPerSec in v2.8.2.tiered)	叢集到達 ACTIVE 狀態之後。	讀取請求之回應的總錯誤率。此類請求是由指定的代理程式傳送至分層儲存以擷取資料，以回應取用者擷取。此指標包含會產生下游資料傳輸流量的所有主題分區。類別：流量和錯誤率。這是一個 KIP-405 指標。

名稱	可見時	描述
RemoteFetchRequestsPerSec (RemoteReadRequestsPerSec in v2.8.2.tiered)	叢集到達 ACTIVE 狀態之後。	讀取請求的總數。此類請求是由指定的代理程式傳送至分層儲存以擷取資料，以回應取用者擷取。此指標包括產生下游資料傳輸流量的所有主題分區。類別：流量和錯誤率。這是一個 KIP-405 指標。
RemoteCopyErrorsPerSec (RemoteWriteErrorPerSec in v2.8.2.tiered)	叢集到達 ACTIVE 狀態之後。	寫入請求之回應的總錯誤率。此類請求是由指定的代理程式傳送至分層儲存以向上游傳輸資料。此指標包括會產生上游資料傳輸流量的所有主題分區。類別：流量和錯誤率。這是一個 KIP-405 指標。
ReplicationBytesInPerSec	建立主題之後。	從其他代理程式接收的每秒位元組數。
ReplicationBytesOutPerSec	建立主題之後。	每秒傳送給其他代理程式的位元組數。
RequestExemptFromThrottleTime	套用請求調節之後。	在代理程式網路和 I/O 執行緒間處理免除調節的請求所花費的平均時間 (毫秒)。
RequestHandlerAvgIdlePercent	叢集到達 ACTIVE 狀態之後。	請求處理常式執行緒閒置的平均時間百分比。
RequestThrottleQueueSize	套用請求調節之後。	調節佇列中的訊息數量。
RequestThrottleTime	套用請求調節之後。	平均請求調節時間 (以毫秒為單位)。
TcpConnections	叢集到達 ACTIVE 狀態之後。	顯示已設定 SYN 旗標的傳入和傳出 TCP 區段數。

名稱	可見時	描述
RemoteCopyLagBytes (TotalTierBytesLag in v2.8.2.tiered)	建立主題之後。	有資格在代理程式上進行分層儲存，但尚未傳輸至分層儲存的資料位元組總數。此指標顯示上游資料傳輸的效率。隨著延遲增加，不會持續存在於分層儲存中的資料量也會增加。類別：存檔延遲。這不是一個 KIP-405 指標。
TrafficBytes	叢集到達 ACTIVE 狀態之後。	顯示用戶端 (生產者和取用者) 與代理程式之間的總網路流量 (位元組)。不報告代理程式之間的流量。
VolumeQueueLength	叢集到達 ACTIVE 狀態之後。	指定期間內等待完成的讀取與寫入操作請求總數。
VolumeReadBytes	叢集到達 ACTIVE 狀態之後。	指定期間內讀取的位元組數。
VolumeReadOps	叢集到達 ACTIVE 狀態之後。	指定期間內的讀取操作數。
VolumeTotalReadTime	叢集到達 ACTIVE 狀態之後。	指定期間內完成之所有讀取操作耗用的總秒數。
VolumeTotalWriteTime	叢集到達 ACTIVE 狀態之後。	指定期間內完成之所有寫入操作耗用的總秒數。
VolumeWriteBytes	叢集到達 ACTIVE 狀態之後。	指定期間內寫入的位元組數。
VolumeWriteOps	叢集到達 ACTIVE 狀態之後。	指定期間內的寫入操作數。

PER_TOPIC_PER_BROKER 層級監控

將監控層次設定為 PER_TOPIC_PER_BROKER 時，除了所有 PER_BROKER 和 DEFAULT 層集的所有指標以外，還會取得下列表格描述的指標。只有 DEFAULT 層級指標是免費的。此表格中的指標包含下列維度：Cluster Name (叢集名稱)、Broker ID (代理程式 ID)、Topic (主題)。

> **Important**
>
> 針對使用 Apache Kafka 2.4.1 或更新版本的 Amazon MSK 叢集，下表中的指標只會在其值首次變為非零值時才會出現。例如，如要查看 BytesInPerSec，一或多個生產者必須先將資料傳送到叢集。

從 **PER_TOPIC_PER_BROKER** 監控層級開始可額外使用的指標

名稱	可見時	描述
FetchMessageConversionsPerSec	建立主題之後。	每秒轉換的擷取訊息數量。
MessagesInPerSec	建立主題之後。	每秒接收的訊息數量。
ProduceMessageConversionsPerSec	建立主題之後。	產生訊息的每秒轉換次數。
RemoteFetchBytesPerSec (RemoteBytesInPerSec in v2.8.2.tiered)	建立主題，並且主題生產/取用資料後。	為回應指定主題和代理程式的取用者擷取而從分層儲存傳輸的位元組數。此指標包含對指定代理程式產生下游資料傳輸流量之主題的所有分區。類別：流量和錯誤率。這是一個 KIP-405 指標。
RemoteCopyBytesPerSec (RemoteBytesOutPerSec in v2.8.2.tiered)	建立主題，並且主題生產/取用資料後。	指定主題和代理程式傳輸至分層儲存的位元組數。此指標包括對指定代理程式產生上游資料傳輸流量之主題的所有分區。類別：流量和錯誤率。這是一個 KIP-405 指標。
RemoteFetchErrorsPerSec (RemoteReadErrorPerSec in v2.8.2.tiered)	建立主題，並且主題生產/取用資料後。	讀取請求之回應的錯誤率。此類請求是由指定的代理程式傳送至分層儲存以擷取資料，以回應對特定主題的取用者擷取。此指標包含對指定代理

名稱	可見時	描述
		程式產生下游資料傳輸流量之主題的所有分區。類別：流量和錯誤率。這是一個 KIP-405 指標。
RemoteFetchRequestsPerSec (RemoteReadRequestsPerSec in v2.8.2.tiered)	建立主題，並且主題生產/取用資料後。	讀取請求的數量。此類請求是由指定的代理程式傳送至分層儲存以擷取資料，以回應對特定主題的取用者擷取。此指標包含對指定代理程式產生下游資料傳輸流量之主題的所有分區。類別：流量和錯誤率。這是一個 KIP-405 指標。
RemoteCopyErrorsPerSec (RemoteWriteErrorPerSec in v2.8.2.tiered)	建立主題，並且主題生產/取用資料後。	寫入請求之回應的錯誤率。此類請求是由指定的代理程式傳送至分層儲存以向上游傳輸資料。此指標包括對指定代理程式產生上游資料傳輸流量之主題的所有分區。類別：流量和錯誤率。這是一個 KIP-405 指標。

PER_TOPIC_PER_PARTITION 層級監控

將監控層級設定為 PER_TOPIC_PER_PARTITION 後，除了所有來自 PER_TOPIC_PER_BROKER、PER_BROKER 和 DEFAULT 層集的指標之外，還會取得下列表格內描述的指標。只有 DEFAULT 層級指標是免費的。此表格中的指標具有下列維度：取用者群組、主題、分區。

從 **PER_TOPIC_PER_PARTITION** 監控層級開始可額外使用的指標

名稱	可見時	描述
EstimatedTimeLag	取用者群組取用一個主題之後。	耗盡分區偏移延遲的估計時間 (秒)。
OffsetLag	取用者群組取用一個主題之後。	分區層級取用者延遲 (偏移量)。

檢視 Amazon MSK 指標，使用 CloudWatch

您可以使用 CloudWatch 主控台、命令列或 CloudWatch API 監控 Amazon MSK 的指標。以下程序將說明如何使用這些不同的方法來存取指標。

使用 CloudWatch 主控台存取指標

請登入 AWS Management Console 並開啟 CloudWatch 主控台，網址為 https://console.aws.amazon.com/cloudwatch/。

1. 在導覽窗格中，選擇 指標。
2. 選擇所有指標索引標籤，然後選擇 AWS/Kafka。
3. 若要檢視主題層級指標，請選擇 Topic, Broker ID, Cluster Name (主題、代理程式 ID、叢集名稱)；針對代理程式層級指標，請選擇 Broker ID, Cluster (代理程式 ID、叢集名稱)；對於叢集層級指標，請選擇 Cluster Name (叢集名稱)。
4. (選擇性) 在圖表窗格中，選取統計值和期間，然後使用這些設定建立 CloudWatch 警示。

若要使用存取量度 AWS CLI

使用 list-metrics 和 get-metric-statistics 命令。

若要使用 CloudWatch CLI 存取指標

使用 mon-list-metrics 和 mon-get-stats 命令。

若要使用 CloudWatch API 存取指標

使用ListMetrics和GetMetric統計資料作業。

取用者延遲監控

監控取用者延遲使您可以識別緩慢或停滯的取用者，他們無法跟上主題中可用的最新資料。必要時，您可以採取補救措施，例如擴展或重新啟動這些取用者。要監控消費者滯後，您可以使用 Amazon CloudWatch 或使用 Prometheus 開放監控。

取用者延遲指標可量化寫入主題的最新資料，與應用程式讀取的資料之間的差異。Amazon MSK 提供下列消費者滯後指標，您可以透過 Amazon CloudWatch 或透過 Prometheus 開放式監控取得這些指標：`EstimatedMaxTimeLag`、、、和。`EstimatedTimeLag` `MaxOffsetLag` `OffsetLag` `SumOffsetLag`如需這些指標的相關資訊，請參閱 the section called “用於監控的 Amazon MSK 指標 CloudWatch”。

> **Note**
>
> 只有處於穩定狀態的消費者群體才能看到消費者滯後指標。在成功完成重新平衡之後，消費者群組是穩定的，確保分割區在消費者之間均勻分佈。

Amazon MSK 支援針對採用 Apache Kafka 2.2.1 及更高版本的叢集使用取用者延遲指標。

使用 Prometheus 進行開放式監控

您可以使用 Prometheus 監控您的 MSK 叢集，前者是一種適用於時間序列指標資料的開放原始碼監控系統。您可以使用 Prometheus 的遠端寫入功能，將此資料發布到適用 Amazon Managed Service for Prometheus。您也可以使用與 Prometheus 格式指標相容的工具，或是與 Amazon MSK 開放式監控整合的工具，例如 Datadog、Lenses、New Relic，以及 Sumo logic。開放式監控提供免費使用，但是針對跨可用區域的資料傳輸則會產生費用。如需 Prometheus 的資訊，請參閱 Prometheus documentation。

建立啟用開放式監控的 Amazon MSK 叢集

使用 AWS Management Console

1. 登入 AWS Management Console，然後開啟 Amazon MSK 主控台，網址為 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。
2. 在監控區段中，選取使用 Prometheus 啟用開放式監控旁邊的核取方塊。
3. 在頁面的所有區段內提供必要資訊，然後檢閱所有可用選項。
4. 選擇建立叢集。

使用 AWS CLI

- 呼叫 create-cluster 命令，並指定其 `open-monitoring` 選項。啟用 `JmxExporter`、`NodeExporter`，或同時啟用兩者。如果您指定 `open-monitoring`，則無法同時停用兩個匯出工具。

使用 API

- 呼叫作CreateCluster業並指定`OpenMonitoring`。啟用 `jmxExporter`、`nodeExporter`，或同時啟用兩者。如果您指定 `OpenMonitoring`，則無法同時停用兩個匯出工具。

為現有的 Amazon MSK 叢集啟用開放式監控

若要啟用開放式監視，請確定叢集處於 ACTIVE 狀態。

使用 AWS Management Console

1. 登入 AWS Management Console，然後開啟 Amazon MSK 主控台，網址為 https://console.aws.amazon.com/msk/home?region=us-east-1#/home/。
2. 選擇您要更新的叢集名稱。這會帶您前往包含叢集詳細資訊的頁面。
3. 在屬性標籤上，向下捲動並找到監控區段。
4. 選擇編輯。
5. 選取使用 Prometheus 啟用開放式監控旁邊的核取方塊。
6. 選擇儲存變更。

使用 AWS CLI

- 呼叫 update-monitoring 命令，並指定其 open-monitoring 選項。啟用 JmxExporter、NodeExporter，或同時啟用兩者。如果您指定 open-monitoring，則無法同時停用兩個匯出工具。

使用 API

- 呼叫作UpdateMonitoring業並指定OpenMonitoring。啟用 jmxExporter、nodeExporter，或同時啟用兩者。如果您指定 OpenMonitoring，則無法同時停用兩個匯出工具。

在 Amazon EC2 執行個體上設定 Prometheus 主機

1. 從 https://prometheus.io/download/#prometheus 將 Prometheus 伺服器下載到您的 Amazon EC2 執行個體。
2. 將下載的檔案解壓縮到目錄，並前往該目錄。
3. 使用下列內容建立名為 prometheus.yml 的檔案。

```
# 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. 使用此ListNodes作業取得叢集代理程式的清單。
5. 使用以下 JSON 建立名為 `targets.json` 的檔案。將 *broker_dns_1* 取代成 *broker_dns_2*，並將剩餘的代理程式 DNS 名稱取代成您在上一個步驟中為您代理程式取得的 DNS 名稱。包括您在上一個步驟中獲得的所有代理程式。Amazon MSK 會將連接埠 11001 用於 JMX Exporter，將連接埠 11002 用於 Node Exporter。

   ZooKeeper mode targets.json

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

```
      "broker_dns_2:11002",
      .
      .
      .
      "broker_dns_N:11002"
    ]
  }
]
```

KRaft mode targets.json

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

> **Note**
>
> 要從 Kraft 控制器抓取 JMX 指標，請在 JSON 文件中添加控制器 DNS 名稱作為目標。例如：`controller_dns_1:11001controller_dns_1`以實際控制器 DNS 名稱取代。

6. 若要在您的 Amazon EC2 執行個體上啟動 Prometheus 伺服器，請在您解壓縮 Prometheus 檔案及儲存 `prometheus.yml` 和 `targets.json` 的目錄中執行以下命令。

```
./prometheus
```

7. 尋找您在上一個步驟中執行 Prometheus 的 Amazon EC2 執行個體 IPv4 公有 IP 地址。您在下一個步驟中需要此公有 IP 地址。
8. 如要存取 Prometheus Web UI，請開啟可存取您 Amazon EC2 執行個體的瀏覽器，然後前往 *`Prometheus-Instance-Public-IP`*`:9090`，其中 *`Prometheus-Instance-Public-IP`* 是您在上一個步驟中取得的公有 IP 地址。

Prometheus 指標

所有由 Apache Kafka 發到 JMX 的指標都可以透過 Prometheus，使用開放式監控存取。如需 Apache Kafka 指標的資訊，請參閱 Apache Kafka 文件中的 Monitoring。除了 Apache Kafka 指標，、名為 `kafka.consumer.group:type=ConsumerLagMetrics` 的 JMX MBean 下的連接埠 11001 還會提供取用者延遲指標。您也可以使用 Prometheus Node Exporter，為您在端口 11002 的代理程式獲取 CPU 和磁盤指標。

將 Prometheus 指標存放在 Amazon Managed Service for Prometheus

Amazon Managed Service for Prometheus 是與 Prometheus 相容的監控和警示服務，可讓您用來監控 Amazon MSK 叢集。這是一項全受管服務，既可自動擴展指標的擷取、儲存、查詢和提醒，它還與 AWS 安全服務集成，為您提供快速安全的數據訪問。您可以使用開放原始碼 PromQL 查詢語言來查詢指標並根據指標發出提醒。

如需詳細資訊，請參閱《Amazon Managed Service for Prometheus 入門》。

Amazon MSK 儲存容量警示

在 Amazon MSK 佈建的叢集上，您可以選擇叢集的主要儲存容量。如果您耗盡已佈建叢集中代理程式的儲存容量，其可能會影響其產生和使用資料的能力，進而導致昂貴的停機時間。Amazon MSK 提供

的 CloudWatch指標可協助您監控叢集的儲存容量。不過，為了讓您更輕鬆地偵測並解決儲存容量問題，Amazon MSK 會自動傳送動態叢集儲存容量警示給您。儲存容量警示包含管理叢集儲存容量相關的短期和長期步驟建議。在 Amazon MSK 主控台中，您可以使用警示中的快速連結立即採取建議的動作。

MSK 儲存容量警示有兩種類型：主動式和補救式。

- 主動式 (「需執行動作」) 儲存容量警示會警告您叢集可能發生的儲存問題。當 MSK 叢集中的代理程式使用其磁碟儲存容量的 60% 或 80% 以上時，受影響的代理程式將主動警示您。
- 當 MSK 叢集中的其中一個代理程式已耗盡磁碟儲存容量時，補救式 (「需執行關鍵動作」) 儲存容量警示會要求您採取補救措施來修正重大叢集問題。

Amazon MSK 會自動將這些警示傳送到 Amazon MSK 主控台、AWS Health 儀表板 EventBridge、Amazon 和您 AWS 帳戶的電子郵件聯絡人。您也可以 EventBridge將 Amazon 設定為將這些警示傳送至 Slack 或新文物和資料多等工具。

所有 MSK 佈建的叢集都已預設啟用儲存容量警示，且無法關閉。提供 MSK 的所有地區都支援此功能。

監控 Amazon MSK 儲存容量警示

您可以透過多種方式查看儲存容量警示：

- 前往 Amazon MSK 主控台。儲存容量警示會在叢集警示窗格中顯示 90 天。警示內包含可解決磁碟儲存容量問題的建議和一鍵式連結，按一下連結即可採取動作。
- 使用ListClustersDescribeCluster、ListClustersV2 或 DescribeClusterV2 API 來檢視叢集`CustomerActionStatus`和所有警示。
- 移至AWS 健全狀況儀表板以檢視來自 MSK 和其他 AWS 服務的警示。
- 設定 AWS Health API 和 Amazon， EventBridge將警示通知路由到第三方平台， NewRelic例如 Datadog 和 Slack。

使用 LinkedIn的巡航控制阿帕奇卡夫卡與 Amazon MSK

您可以使用 LinkedIn巡航控制重新平衡 Amazon MSK 叢集、偵測和修復異常情況，以及監控叢集的狀態和運作狀態。

下載並建立 Cruise Control

1. 在與 Amazon MSK 叢集相同的 Amazon VPC 中建立 Amazon EC2 執行個體。
2. 在您於前一個步驟中建立的 Amazon EC2 執行個體上安裝 Prometheus。請記下私有 IP 和連接埠。預設連接埠號碼為 9090。如需有關如何設定 Prometheus 來彙整您叢集指標的詳細資訊，請參閱 the section called “使用 Prometheus 進行開放式監控”。
3. 在 Amazon EC2 執行個體上下載 Cruise Control。(或者若您偏好，也可以僅將 Cruise Control 用於個別的 Amazon EC2 執行個體。) 若為使用 Apache Kafka 2.4.* 版的叢集，請使用最新的 2.4.* Cruise Control 版。若您叢集的 Apache Kafka 版本低於 2.4.*，請使用最新的 2.0.* Cruise Control 版。
4. 解壓縮 Cruise Control 檔案，然後前往已解壓縮的文件夾。
5. 執行以下命令來安裝 git。

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

6. 執行以下命令來初始化本機儲存庫。使用您目前資料夾 (解壓縮 Cruise Control 下載內容時獲得的資料夾) 的名稱來取代 *Your-Cruise-Control-Folder*。

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

7. 執行以下命令來建置原始碼。

```
./gradlew jar copyDependantLibs
```

設定和執行 Cruise Control

1. 對 `config/cruisecontrol.properties` 檔案進行以下更新。將示例啟動程序服務器和引導程序代理程序字符串替換為集群的值。若要取得叢集的這些字串，請在主控台中查看叢集詳細資料。或者，您可以使用GetBootstrapBrokers和 DescribeClusterAPI 作業或其 CLI 對等項目。

```
# If using TLS encryption, use 9094; use 9092 if using plaintext
```

```
bootstrap.servers=b-1.test-cluster.2skv42.c1.kafka.us-
east-1.amazonaws.com:9094,b-2.test-cluster.2skv42.c1.kafka.us-
east-1.amazonaws.com:9094,b-3.test-cluster.2skv42.c1.kafka.us-
east-1.amazonaws.com:9094

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

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

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

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

2. 編輯 `config/capacityCores.json` 檔案以指定正確的磁碟大小和 CPU 核心數，以及網路輸入/輸出限制。您可以使用 DescribeClusterAPI 作業 (或其 CLI 等效項目) 來取得磁碟大小。如需有關 CPU 核心數和網路輸入/輸出限制的詳細資訊，請參閱 Amazon EC2 執行個體類型。

```
{
  "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. 您可選擇安裝 Cruise Control UI。若要下載，請前往 Setting Up Cruise Control Frontend。
4. 執行以下命令以啟動 Cruise Control。請考慮使用類似 `screen` 或 `tmux` 之工具以持續開啟長時間執行的工作階段。

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

5. 使用 Cruise Control API 或 UI，以確保 Cruise Control 具有叢集負載資料，且正在提出重新平衡建議。可能需要花費幾分鐘來取得指標的有效視窗。

Amazon MSK 巡航控制的自動化部署範本

您也可以使用此CloudFormation 範本輕鬆部署巡航控制和 Prometheus，以深入瞭解 Amazon MSK 叢集的效能並最佳化資源使用率。

主要特色

- 使用巡航控制和 Prometheus 預先設定的自動佈建 Amazon EC2 執行個體。
- Support Amazon MSK 佈建的叢集。
- 靈活的身份驗證PlainText 與 IAM。
- 沒有動物園管理員的依賴巡航控制。
- 透過提供存放在 Amazon S3 儲存貯體中的自己的組態檔案，輕鬆自訂 Prometheus 目標、巡航控制容量設定和其他組態。

Amazon MSK 配額

您的 AWS 帳戶具有 Amazon MSK 的預設配額。除非另有說明，否則每個帳戶的每個配額都是您帳戶中的區域特定。 AWS

Amazon MSK 配額

- 每個模式集群 30 個經紀人。每個 Kraft ZooKeeper 模式集群 60 經紀人。要請求更高的配額，請前往主 AWS 控台 Support 中心並建立支援案例。
- 每個代理程式至少 1 GiB 的儲存空間。
- 每個代理程式最多 16384 GiB 的儲存空間。
- 在任何時間，使用 the section called “IAM 存取控制” 的叢集，其每個代理程式最多可有 3000 個 TCP 連線。若要增加此限制，您可以使用 Kafka AlterConfig API `listener.name.client_iam.max.connections` 或工具調整或`listener.name.client_iam_public.max.connections`組態屬性。`kafka-configs.sh`請務必注意，將任一屬性值提高可能會導致無法使用。
- TCP 連線的限制。啟用連線速率突發時，MSK 允許每秒 100 個連線。kafka.t3.small 執行個體類型為例外，允許每秒 4 個連線，且啟用連線速率突增。未啟用連線速率突發的較舊叢集會在修補叢集時自動啟用此功能。

  若要處理失敗連線的重試，您可在用戶端設定 `reconnect.backoff.ms` 組態參數。例如，若您要讓用戶端在 1 秒後重試連線，請將 `reconnect.backoff.ms` 設為 1000。如需詳細資訊，請參閱 Apache Kafka 文件中的 reconnect.backoff.ms。
- 每個帳戶最多 100 個組態。若要請求調整配額，請前往 AWS 主控台支援中心，然後建立支援案例。
- 每個組態最多 50 個修訂版本。
- 若要更新 MSK 叢集的組態或 Apache Kafka 版本，請先確認每個代理程式的分區數量未超過 the section called “ 適當調整叢集大小：每個代理程式的分區數量” 中所述的限制。

MSK Replicator 配額

- 每個帳戶最多 15 個 MSK Replicator。

- MSK 複製器最多只能以排序順序複製 750 個主題。如果您需要複製更多主題，建議您建立個別的複寫器。如果您需要每個複製器超過 750 個主題的支援，請前往主 AWS 控台支援中心並建立支援案例。您可以使用 "TopicCount" 測量結果監督複製的主題數目。
- 每個 MSK Replicator 的傳入輸送量上限為每秒 1 GB。要請求更高的配額，請前往主 AWS 控台 Support 中心並建立支援案例。
- MSK 複製器記錄大小-最大 10MB 的記錄大小 (訊息. 最大位元組)。要請求更高的配額，請前往主 AWS 控台 Support 中心並建立支援案例。

MSK Serverless 配額

> **Note**
>
> 如果您在配額限制方面遇到任何問題，請建立 Sup AWS port 案例與支援部門聯絡。

除非另有說明，否則限制的適用單位為每個叢集。

維度	配額	配額違反結果
傳入輸送量上限	200 MBps	回應變慢，遭遇限流
傳出輸送量上限	400 MBps	回應變慢，遭遇限流
保留期間上限	無限制	N/A
用戶端連線數上限	3000	連線關閉
連線嘗試次數上限	每秒 100 次	連線關閉
訊息大小上限	8 MB	請求失敗，顯示 ErrorCode：無效請求
請求速率上限	每秒 15,000 個	回應變慢，遭遇限流
主題管理 API 請求速率上限	每秒 2 個	回應變慢，遭遇限流
每個請求的擷取位元組數上限	55 MB	請求失敗，顯示 ErrorCode：無效請求

維度	配額	配額違反結果
取用者群組數上限	500	JoinGroup 請求失敗
最大分割區數目 (引線)	非壓縮主題為 2400 個；壓縮主題為 120 個。若要申請配額調整，請前往 AWS 主控台 Support 中心並建立支援案例。	請求失敗，顯示 ErrorCode：無效請求
分區建立和刪除速率上限	5 分鐘 250 個	請求失敗，顯示 ErrorCode：超過輸送量
每個分區的傳入輸送量上限	5 MBps	回應變慢，遭遇限流
每個分區的傳出輸送量上限	10 MBps	回應變慢，遭遇限流
分區大小上限 (適用於壓縮主題)	250 GB	請求失敗，顯示 ErrorCode：超過輸送量
每個無伺服器叢集的用戶端 VPC 數量上限	5	
每個帳戶的無伺服器叢集數量上限	10. 若要申請配額調整，請前往 AWS 主控台 Support 中心並建立支援案例。	

MSK Connect 配額

- 最高 100 個自訂外掛程式。
- 最高 100 個工作程序組態。
- 最高 60 個連線工作程序。若連接器設定為具有自動擴展容量，則連接器設定擁有的工作程序數量上限會是 MSK Connect 用來計算帳戶配額的數量。
- 每個連接器最多 10 個工作程序。

若要要求更高的 MSK Connect 配額，請前往 AWS 主控台 Support 中心並建立支援案例。

Amazon MSK 資源

資源一詞在 Amazon MSK 中具有兩個含義，具體取決於內容。在 API 內容中，資源是您可以在其中調用操作的結構。如需這些資源的清單以及您可以在這些資源上調用之操作道清單，請參閱 Amazon MSK API Reference 中的 Resources。在 the section called “IAM 存取控制” 內容中，資源是您可以允許或拒絕存取的實體，如 the section called “資源” 章節所定義。

MSK 整合

本節提供與 Amazon MSK 整合之 AWS 功能的參考資料。

主題

適用於 Amazon MSK 的 Amazon Athena 連接器

適用於 Amazon MSK 的 Amazon Athena 連接器可讓 Amazon Athena 能夠對 Apache Kafka 主題執行 SQL 查詢。使用此連接器可以在 Athena 中以資料表的形式檢視 Apache Kafka 主題，並以資料列的形式檢視訊息。

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的 Amazon Athena MSK 連接器。

Amazon Redshift 串流資料擷取

Amazon Redshift 支援從 Amazon MSK 串流擷取。Amazon Redshift 串流擷取功能可實現低延遲、高速將串流資料從 Amazon MSK 擷取到 Amazon Redshift 具體化視觀表。由於無需在 Amazon S3 中暫存資料，因此 Amazon Redshift 可採用較低的延遲和更低的儲存成本擷取串流資料。您可以使用 SQL 陳述式在 Amazon Redshift 叢集上設定 Amazon Redshift 串流擷取，以進行身分驗證並連線至 Amazon MSK 主題。

如需詳細資訊，請參閱 Amazon Redshift Database Developer Guide 中的 Streaming ingestion。

Firehose

Amazon MSK 與 Firehose 整合，提供無伺服器、無程式碼的解決方案，可將串流從 Apache Kafka 叢集交付到 Amazon S3 資料湖。Firehose 是一種串流擷取、轉換和載入 (ETL) 服務，可從您的 Amazon MSK Kafka 主題讀取資料、執行轉換 (例如轉換到實木地板)，以及將資料彙總並寫入 Amazon S3。只需從主控台按幾下，您就可以設定 Firehose 串流，從 Kafka 主題讀取並交付到 S3 位置。無需撰寫程式碼、無需連接器應用程式，也沒有要佈建的資源。Firehose 會根據發佈到 Kafka 主題的資料量自動調整規模，而且您只需支付從 Kafka 擷取的位元組付費。

如需有關此功能的詳細資訊，請參閱以下內容。

- 使用 Amazon MSK 寫入 Kinesis Data Firehose-亞馬 Amazon Kinesis Data Firehose 開發人員指南中的 Amazon Kinesis 資料防火軟管
- 部落格：Amazon MSK Introduces Managed Data Delivery from Apache Kafka to Your Data Lake
- 實驗室：使用 Firehose 交付到 Amazon S3

通過 Amazon MSK 控制台訪問 Amazon EventBridge 管道

Amazon EventBridge 管道將源連接到目標。管道旨在用於支持的源和目標之間的 point-to-point集成，並支持高級轉換和擴展。 EventBridge 管道提供可高度擴展的方式，將 Amazon MSK 叢集連接到 Step Functions、Amazon SQS 和 API Gateway 等 AWS 服務，以及第三方軟體即服務 (SaaS) 應用程式 (例如 Salesforce)。

若要設定管道，您可以選擇來源、新增可選篩選、定義可選的擴充，以及選擇事件資料的目標。

在 Amazon MSK 叢集的詳細資料頁面上，可以檢視使用該叢集作為其來源的管道。從那裡，您還可以：

- 啟動 EventBridge 控制台以查看管道詳細信息。
- 啟動主 EventBridge 控台以建立新管道，並將叢集做為其來源。

如需將 Amazon MSK 叢集設定為管道來源的詳細資訊，請參閱 Amazon 使用者指南中的 Amazon Managed Streaming for Apache Kafka 作為來源。 EventBridge 若要取得有關一般 EventBridge 管的更多資訊，請參閱〈EventBridge 管〉。

存取指定 Amazon MSK 叢集的 EventBridge 管道

1. 開啟 Amazon MSK 主控台並選擇叢集。
2. 選取叢集。
3. 在叢集詳細資訊頁面中，選擇整合索引標籤。

   整合索引標籤包括目前設定為使用所選叢集作為來源的所有管道的清單，包括：

   - 管道名稱
   - 目前的狀態
   - 管道目標

- 上次修改管道的時間

4. 視需要管理 Amazon MSK 叢集的管道：

   存取有關管道的更多詳細資訊

   - 選擇管道。

     這將啟動 EventBridge 控制台的管道詳細信息頁面。

   建立新的管道

   - 選擇將 Amazon MSK 叢集連接至管道。

     這會啟動 EventBridge 主控台的「建立管道」頁面，並將 Amazon MSK 叢集指定為管道來源。如需詳細資訊，請參閱 Amazon EventBridge 使用者指南中的建立 EventBridge 管道。

   - 您也可以從叢集頁面上建立叢集的管道。選取叢集，然後從「動作」功能表中選取「建立 EventBridge 管道」。

Apache Kafka 版本

建立 Amazon MSK 叢集時，可指定要在其上使用的 Apache Kafka 版本。您也可以更新現有叢集的 Apache Kafka 版本。本章中的主題可協助您瞭解 Kafka 版本支援的時間表，以及最佳做法的建議。

主題

支援的 Apache Kafka 版本

Amazon Managed Streaming for Apache Kafka (Amazon MSK) 支援以下的 Apache Kafka 和 Amazon MSK 版本。Apache 卡夫卡社群在發行日期後提供大約 12 個月的版本支援。欲了解更多詳細信息，請檢查阿帕奇卡夫卡 EOL（生命週期結束）政策。

支援的 Apache Kafka 版本

阿帕奇卡夫卡版	MSK 發行日期	支援結束日期
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

阿帕奇卡夫卡版	MSK 發行日期	支援結束日期
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
分層	2022-10-28	即將公佈
3.1.1	2022-06-22	2024-09-11
3.2.0	2022-06-22	2024-09-11
3.3.1	2022-10-26	2024-09-11
3.3.2	2023-03-02	2024-09-11
3.4.0	2023-05-04	2025-06-17
3.5.1 (建議使用)	2023-09-26	--
3.6.0	2023-11-16	--
3.7.x	2024-05-29	--

如需 Amazon MSK 版本支援政策的詳細資訊，請參閱Amazon MSK 版本支援政策。

阿帕奇卡夫卡 3.7.x 版（與生產就緒分層存儲）

阿帕奇卡夫卡 3.7.x 版本在 MSK 包括阿帕奇卡夫卡版本 3.7.0 的支持。您可以建立叢集或升級現有叢集以使用新的 3.7.x 版本。隨著版本命名的變更，您不再需要在 Apache Kafka 社群發行時採用更新的修補程式修正版本，例如 3.7.1。Amazon MSK 將自動更新 3.7.x，以支援 future 的修補程式版本可供使用。這可讓您受益於透過修補程式修正版本提供的安全性和錯誤修正，而不會觸發版本升級。Apache Kafka 發行的這些修補程式修正版本不會破壞版本相容性，您可以從新的修補程式修正版

本中受益，而不必擔心用戶端應用程式的讀取或寫入錯誤。請確定您的基礎架構自動化工具 (例如) 已更新 CloudFormation，以說明版本命名中的這項變更。

Amazon MSK 現在支持卡夫卡模式（阿帕奇卡夫卡筏）在阿帕奇卡夫卡 3.7.x 版。在 Amazon MSK 上，就像 ZooKeeper 節點一樣，Kraft 控制器隨附在內，無需額外付費，而且不需要額外的設定或管理。現在，您可以在 Apache 卡夫卡 3.7.x 版本上創建卡夫卡 ZooKeeper 模式或模式集群。在 Kraft 模式下，與 Zookeeper 型叢集上的 30 個代理程式配額相比，您最多可以新增 60 個代理程式來裝載每個叢集更多的磁碟分割，而不需要增加限制。要了解有關 MSK 卡夫的更多信息，請參閱 Kraft 模式。

阿帕奇卡夫卡 3.7.x 版還包括幾個錯誤修復和新功能，以提高性能。主要改進包括針對用戶端的領導者探索最佳化和記錄區段清除最佳化選項。如需改善和錯誤修正的完整清單，請參閱 Apache Kafka 3.7.0 版本說明。

Apache Kafka 3.6.0 版本 (具有已準備好投入生產的分層儲存)

如需有關 Apache Kafka 3.6.0 版本 (具有已準備好投入生產的分層儲存) 的資訊，請參閱 Apache Kafka 下載網站上的版本備註。

Amazon MSK 將在此版本中繼續使用和管理 Zookeeper 來進行規定人數管理，以確保穩定性。

Amazon MSK 3.5.1 版

Amazon 阿帕奇卡夫卡（Amazon MSK）受管流媒體現在支持 Apache 卡夫卡 3.5.1 版用於新的和現有的集群。阿帕奇卡夫卡 3.5.1 包括幾個錯誤修復和新功能，以提高性能。主要功能包括為消費者引入新的機架感知分割區指派。Amazon MSK 將在此版本中繼續使用和管理動物園管理員進行仲裁管理。如需改善和錯誤修正的完整清單，請參閱 Apache Kafka 3.5.1 版本說明。

如需有關 Apache Kafka 3.5.1 版的資訊，請參閱 Apache Kafka 下載網站上的版本備註。

Amazon MSK 版本 3.4.0

Amazon 阿帕奇卡夫卡（Amazon MSK）受管流媒體現在支持 Apache 卡夫卡 3.4.0 版本適用於新的和現有的集群。阿帕奇卡夫卡 3.4.0 包括幾個錯誤修復和新功能，可以提高性能。主要功能包括修復程序，以提高從最近的副本獲取的穩定性。Amazon MSK 將在此版本中繼續使用和管理動物園管理員進行仲裁管理。如需改善和錯誤修正的完整清單，請參閱 Apache Kafka 3.4.0 版本說明。

如需有關 Apache Kafka 3.4.0 版的資訊，請參閱 Apache Kafka 下載網站上的版本備註。

Amazon MSK 3.3.2 版

Amazon 阿帕奇卡夫卡（Amazon MSK）受管流媒體現在支持 Apache 卡夫卡 3.3.2 版用於新的和現有的集群。阿帕奇卡夫卡 3.3.2 包括幾個錯誤修復和新功能，以提高性能。主要功能包括修復程序，以提高從最近的副本獲取的穩定性。Amazon MSK 將在此版本中繼續使用和管理動物園管理員進行仲裁管理。如需改善和錯誤修正的完整清單，請參閱 Apache Kafka 3.3.2 版本說明。

如需有關 Apache Kafka 3.3.2 版的資訊，請參閱 Apache Kafka 下載網站上的版本備註。

Amazon MSK 版本 3.3.1

Amazon 阿帕奇卡夫卡（Amazon MSK）受管流媒體現在支持 Apache 卡夫卡 3.3.1 版本適用於新的和現有的集群。阿帕奇卡夫卡 3.3.1 包括幾個錯誤修復和新功能，以提高性能。一些關鍵功能包括量度和分區程序的增強功能。Amazon MSK 將在此版本中繼續使用和管理 Zookeeper 來進行規定人數管理，以確保穩定性。如需改善和錯誤修正的完整清單，請參閱 Apache Kafka 3.3.1 版本說明。

如需有關 Apache Kafka 3.3.1 版的資訊，請參閱 Apache Kafka 下載網站上的版本備註。

Amazon MSK 3.1.1 版

Amazon 阿帕奇卡夫卡（Amazon MSK）受管流媒體現在支持 Apache 卡夫卡 3.1.1 和 3.2.0 版本，用於新的和現有的集群。阿帕奇卡夫卡 3.1.1 和阿帕奇卡夫卡 3.2.0 包括幾個錯誤修復和新功能，提高性能。一些關鍵功能包括指標的增強功能和主題 ID 的使用。MSK 將繼續在此版本中使用和管理 Zookeeper 進行法定人數管理，以確保穩定性。如需改善和錯誤修正的完整清單，請參閱 Apache 卡夫卡 3.1.1 和 3.2.0 版本說明。

如需有關阿帕奇卡夫卡 3.1.1 和 3.2.0 版本的資訊，請參閱阿帕奇卡夫卡下載網站上的 3.2.0 發行說明和 3.1.1 版本說明。

Amazon MSK 分層儲存 2.8.2.tiered 版

此版本是 Apache Kafka 2.8.2 版的僅限 Amazon MSK 版本，且與開源 Apache Kafka 用戶端兼容。

2.8.2.tiered 版本包含分層儲存功能，與 Apache Kafka 的 KIP-405 中引入之 API 相容。如需有關 Amazon MSK 分層儲存功能的詳細資訊，請參閱 分層儲存。

Apache Kafka 2.5.1 版

Apache 卡夫卡版本 2.5.1 包括幾個錯誤修復和新功能，包括 Apache ZooKeeper 和管理客戶端的傳輸過程中加密。Amazon MSK 提供 TLS ZooKeeper 端點，您可以透過DescribeCluster 操作進行查詢。

DescribeCluster操作的輸出包括ZookeeperConnectStringTls節點，其中列出了 TLS 動物園管理員端點。

以下範例會顯示 DescribeCluster 操作之回應的 ZookeeperConnectStringTls 節點：

```
"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"
```

如需有關搭配使用 TLS 加密和 ZooKeeper 的相關資訊，請參閱 搭配阿帕奇使用 TLS 安全性 ZooKeeper。

如需有關 Apache Kafka 2.5.1 版的詳細資訊，請參閱 Apache Kafka 下載網站上的版本備註。

Amazon MSK 2.4.1.1 錯誤修正版

此版本是 Apache Kafka 2.4.1 版本的僅限 Amazon MSK 錯誤修正版。此錯誤修正版包含 KAFKA-9752 的修正內容，KAFKA-9752 是一個罕見問題，會造成取用者群組持續重新平衡並維持在 PreparingRebalance 狀態。此問題會影響執行 Apache Kafka 2.3.1 和 2.4.1 版的叢集。此版本包含社群產生的修正內容，適用於 Apache Kafka 2.5.0 版。

> **Note**
>
> 任何與 Apache Kafka 2.4.1 版相容的 Apache Kafka 用戶端，會與執行 2.4.1.1 版的 Amazon MSK 叢集相容。

若您要使用 Apache Kafka 2.4.1 版本，我們建議您針對新的 Amazon MSK 叢集使用 MSK 2.4.1.1 錯誤修正版本。您可以將執行 Apache Kafka 2.4.1 版的現有叢集更新為此版本，以納入此修正內容。如需有關升級現有叢集的資訊，請參閱 更新 Apache Kafka 版本。

若要在不將叢集升級至 2.4.1.1 版的情況下解決此問題，請參閱 疑難排解 Amazon MSK 叢集 指南的 取用者群組停滯在 PreparingRebalance 狀態 章節。

Apache Kafka 2.4.1 版 (改為使用 2.4.1.1 版)

> **Note**
>
> 您無法再使用 Apache Kafka 2.4.1 版來建立 MSK 叢集。您可以改為搭配使用 Amazon MSK 2.4.1.1 錯誤修正版 和與 Apache Kafka Apache 2.4.1 版相容的用戶端。若你已有具有 Apache Kafka 2.4.1 版的 MSK 叢集，建議您進行更新，以改為使用 Apache Kafka 2.4.1.1 版。

KIP-392 是其中一個關鍵的 Kafka 改善提案，包含在 Apache Kafka 2.4.1 版本中。這項改善可讓取用者從最接近的複本擷取。如要使用此功能，請將取用者屬性中的 `client.rack` 設為取用者可用區域的 ID。範例 AZ ID 為 `use1-az1`。Amazon MSK 會將 `broker.rack` 設為代理程式可用區域的 ID。您也必須將 `replica.selector.class` 組態屬性設為 `org.apache.kafka.common.replica.RackAwareReplicaSelector`，此為 Apache Kafka 所提供機架意識的一種實作。

當您使用此版本的 Apache Kafka 時，`PER_TOPIC_PER_BROKER` 監控層級中的指標只有在其值首次變為非零值時才會出現。如需此項目的詳細資訊，請參閱the section called "`PER_TOPIC_PER_BROKER` 層級監控"。

如需如何尋找可用區域 ID 的相關資訊，請參閱 AWS Resource Access Manager 使用者指南中的資源適用的 AZ ID。

如需設定組態屬性的資訊，請參閱 組態。

如需 KIP-392 的詳細資訊，請參閱 Confluence 頁面中的 Allow Consumers to Fetch from Closest Replica。

如需 Apache Kafka 2.4.1 版的詳細資訊，請參閱 Apache Kafka 下載網站上的版本備註。

Amazon MSK 版本支持

本主題描述的Amazon MSK 版本支援政策和程序更新 Apache Kafka 版本。如果您要升級 Kafka 版本，請遵循中列出的最佳做法。版本升級的最佳做法

Amazon MSK 版本支援政策

本節說明支援 Amazon MSK 支援的卡夫卡版本的支援政策。

- 支持所有 Kafka 版本，直到支持日期結束為止。如需終止支援日期的詳細資訊，請參閱支援的 Apache Kafka 版本。在支援結束日期之前，將 MSK 叢集升級至建議的 Kafka 版本或更高版本。如需更新 Apache Kafka 版本的詳細資訊，請參閱。更新 Apache Kafka 版本在支援日期結束後使用 Kafka 版本的叢集會自動升級至建議的 Kafka 版本。
- MSK 將逐步淘汰對新建立的叢集的支援，這些叢集使用 Kafka 版本且已發佈的終止支援日期。

更新 Apache Kafka 版本

您可以將現有的 MSK 叢集更新為較新版本的 Apache Kafka。您無法將其更新為較舊的版本。當您更新 MSK 叢集的 Apache Kafka 版本時，請同時檢查用戶端軟體，以確保其版本可讓您使用叢集的新 Apache Kafka 版本的功能。Amazon MSK 只會更新伺服器軟體。它不會更新您的客戶端。

如需如何在更新期間讓叢集變為高可用性的詳細資訊，請參閱 the section called “建置高可用性叢集”。

> **Important**
>
> 您無法為超過 the section called “ 適當調整叢集大小：每個代理程式的分區數量” 中所述限制的 MSK 叢集更新 Apache Kafka 版本。

更新阿帕奇卡夫卡版本使用 AWS Management Console

1. 開啟位於 https://console.aws.amazon.com/msk/ 的 Amazon MSK 主控台。
2. 選擇要更新其 Apache Kafka 版本的 MSK 叢集。
3. 在屬性索引標籤上，選擇 Apache Kafka 版本區段中的升級。

更新阿帕奇卡夫卡版本使用 AWS CLI

1. 執行下列命令，取代*ClusterArn*為建立叢集時取得的 Amazon 資源名稱 (ARN)。若您沒有叢集的 ARN，可透過列出所有叢集來找到該 ARN。如需詳細資訊，請參閱 the section called “列出叢集”。

   ```
   aws kafka get-compatible-kafka-versions --cluster-arn ClusterArn
   ```

   此命令的輸出包含您可以更新叢集的 Apache Kafka 版本清單。輸出如以下範例所示：

   ```
   {
   ```

```
    "CompatibleKafkaVersions": [
        {
            "SourceVersion": "2.2.1",
            "TargetVersions": [
                "2.3.1",
                "2.4.1",
                "2.4.1.1",
                "2.5.1"
            ]
        }
    ]
}
```

2. 執行下列命令，取代*ClusterArn*為建立叢集時取得的 Amazon 資源名稱 (ARN)。若您沒有叢集的 ARN，可透過列出所有叢集來找到該 ARN。如需詳細資訊，請參閱 the section called “列出叢集”。

   將叢集目前版本取代為 *Current-Cluster-Version*。對於，*TargetVersion*您可以從上一個命令的輸出中指定任何目標版本。

   > **Important**
   >
   > 叢集版本不是簡單的整數。若要尋找叢集的目前版本，請使用DescribeCluster作業或描述 AWS CLI 叢集指令。範例版本為KTVPDKIKX0DER。

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

   前一個命令的輸出如以下 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. 若要取得update-cluster-kafka-version作業的結果，請執行下列命令，將 *ClusterOperationArn* 取代為您在命令輸出中取得的 update-cluster-kafka-version ARN。

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

此 describe-cluster-operation 命令的輸出如以下 JSON 範例所示。

```
{
    "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 具有值 UPDATE_IN_PROGRESS，請稍候一段時間，然後再次執行 describe-cluster-operation 命令。操作完成時，OperationState 的值會變成 UPDATE_COMPLETE。由於 Amazon MSK 完成操作所需的時間不盡相同，因此您可能需要重複檢查，直到操作完成為止。

使用 API 更新 Apache Kafka 版本

1. 呼叫GetCompatibleKafkaVersions作業以取得可以更新叢集的 Apache Kafka 版本清單。
2. 調用UpdateClusterKafkaVersion操作將集群更新為兼容的 Apache 卡夫卡版本之一。

版本升級的最佳做法

若要在作為 Kafka 版本升級程序一部分執行的滾動式更新期間確保用戶端連續性，請檢閱用戶端的組態和 Apache Kafka 主題，如下所示：

- 針對雙可用區叢集，將主題複寫係數 (RF) 設定2為最小值，針對三個可用區叢集設定3為的最小值。的 RF 值2可能會導致在修補期間產生離線分割區。
- 將最小同步複本 (minISR) 設定為的RF - 1最大值，以確保分割區複本集可以容許一個複本離線或複寫不足。
- 設定用戶端使用多個代理程式連線字串。如果開始修補支援用戶端 I/O 的特定代理程式，則在用戶端的連接字串中有多個代理程式可允許容錯移轉。如需如何透過多個代理程式取得連線字串的詳細資訊，請參閱取得 Amazon MSK 叢集的啟動程式代理程式。
- 建議您將連線用戶端升級至建議的版本或更高版本，以享受新版本提供的功能。用戶端升級不受 MSK 叢集 Kafka 版本的生命週期結束 (EOL) 日期限制，也不需要在 EOL 日期之前完成。Apache Kafka 提供雙向用戶端相容性原則，可讓較舊的用戶端使用較新的叢集，反之亦然。
- 使用版本 3.x.x 的卡夫卡客戶端可能會帶有以下默認值：和。acks=all enable.idempotence=true acks=all與先前的預設值不同，acks=1並透過確保所有同步處理複本確認產生請求，以提供額外的耐久性。同樣地，的預設值enable.idempotence為先前false。變更enable.idempotence=true為預設值可降低出現重複郵件的可能性。這些變更被視為最佳實務設定，可能會產生少量的額外延遲，這些延遲位於正常效能參數內。

- 建立新的 MSK 叢集時，請使用建議的 Kafka 版本。使用推薦的 Kafka 版本可以讓您從最新的卡夫卡和 MSK 功能中受益。

疑難排解 Amazon MSK 叢集

下列資訊可協助您對 Amazon MSK 叢集可能發生的問題進行疑難排解。您也可以將您的問題張貼到 AWS re:Post。

主題

磁碟區置換造成磁碟飽和，因為複寫過載

在意外的磁碟區硬體故障期間，Amazon MSK 可能會以新的執行個體取代磁碟區。卡夫卡通過從集群中的其他代理複製分區重新填充新的卷。一旦分區被複製並趕上，它們就有資格獲得領導和同步複本 (ISR) 成員資格。

問題

在從磁碟區更換中復原的代理程式中，某些不同大小的分割區可能會比其他分割區重新連線。這可能會有問題，因為這些分區可能會提供來自同一個代理程序的流量，該代理仍在趕上（複製）其他分區。此複寫流量有時會使基礎磁碟區輸送量限制飽和，在預設情況下為每秒 250 MiB。當這種飽和發生時，任何已經趕上的分割區都會受到影響，導致與那些被抓取的分割區共用 ISR 的代理程式（不僅僅是因為遠端 ACK 而導致的前導分割區`acks=all`）的代理程式在叢集中產生延遲。這個問題更常見於具有大小不同的磁碟分割數目較大的較大叢集。

建議

- 若要改善複寫 I/O 狀況，請確定已設定最佳實務執行緒設定。
- 若要降低基礎磁碟區飽和的可能性，請啟用具有較高輸送量的佈建儲存體。對於高輸送量複寫案例，建議使用最小輸送量值 500 Mb/s，但實際需要的值會隨輸送量和使用案例而有所不同。 佈建儲存輸送量。
- 若要將複製壓力降`num.replica.fetchers`至最低，請降低至的預設值2。

取用者群組停滯在 `PreparingRebalance` 狀態

如果您的一個或多個取用者群組停滯在永久再平衡狀態，原因可能是 Apache Kafka 問題 KAFKA-9752，這會影響到 Apache Kafka 2.3.1 和 2.4.1 版。

若要解決此問題，建議您將叢集升級至 Amazon MSK 2.4.1.1 錯誤修正版，其中包含此問題的修正程式。如需有關將現有叢集更新至 Amazon MSK 2.4.1.1 錯誤修正版的相關資訊，請參閱更新 Apache Kafka 版本。

在不將叢集升級至 Amazon MSK 2.4.1.1 錯誤修正版的情況下，解決此問題的因應措施是設定 Kafka 用戶端使用靜態成員通訊協定或是辨識與重新啟動停滯的取用者群組的協調代理程式節點。

實施靜態成員通訊協定

若要在您的用戶端實施靜態成員通訊協定，請執行以下操作：

1. 將 Kafka Consumers 組態的 `group.instance.id` 屬性設定為靜態字串，且其可辨識群組中的取用者。
2. 確保組態的其他執行個體已更新為使用靜態字串。
3. 將變更部署到您的 Kafka 取用者。

如果用戶端組態中的工作階段逾時設定的值足以讓取用者復原，而不過早觸發取用者群組再平衡，則使用靜態成員通訊協定會比較有效。例如，如果您的取用者應用程式可以容忍無法使用 5 分鐘，則工作階段逾時的合理值將是 4 分鐘，而不是 10 秒的預設值。

> **Note**
>
> 使用靜態成員通訊協定只會降低遇到此問題的可能性。即使使用靜態成員通訊協定，您仍可能會遇到此問題。

重新啟動協調代理程式節點

若要重啟協調代理程式節點，請執行下列操作：

1. 使用 `kafka-consumer-groups.sh` 指令辨識群組協調器。
2. 使用 RebootBrokerAPI 動作重新啟動卡住用戶群組的群組協調器。

將代理日誌傳送到 Amazon CloudWatch 日誌時發生錯

當您嘗試設定叢集以將代理程式日誌傳送到 Amazon Lo CloudWatch gs 時，可能會出現兩個例外狀況之一。

如果出現 `InvalidInput.LengthOfCloudWatchResourcePolicyLimitExceeded` 例外狀況，請再試一次，但請使用開頭為 `/aws/vendedlogs/` 的日誌群組。如需詳細資訊，請參閱啟用從特定 Amazon Web Services 記錄日誌。

如果您收到`InvalidInput.NumberOfCloudWatchResourcePoliciesLimitExceeded`例外狀況，請在您的帳戶中選擇現有的 Amazon CloudWatch 日誌政策，並在其中附加下列 JSON。

```
{"Sid":"AWSLogDeliveryWrite","Effect":"Allow","Principal":
{"Service":"delivery.logs.amazonaws.com"},"Action":
["logs:CreateLogStream","logs:PutLogEvents"],"Resource":["*"]}
```

如果您嘗試將上述 JSON 附加到現有政策，但出現錯誤訊息表示您已達到所選政策的最大長度，請嘗試將 JSON 附加到另一個 Amazon CloudWatch 日誌政策。將 JSON 附加到現有政策後，請再次嘗試將代理程式日誌交付設定到 Amazon CloudWatch 日誌。

沒有預設安全群組

如果您嘗試建立叢集並收到錯誤，指出沒有預設安全群組，可能是因為您正使用與您共用的 VPC。請您的系統管理員授與您描述此 VPC 上安全群組的權限，然後再試一次。如需關於允許此動作的政策範例，請參閱 Amazon EC2：允許以程式設計方式和主控台中管理與特定 VPC 關聯的 EC2 安全群組。

叢集顯示停滯在 CREATING 狀態

有時叢集建立最多需要 30 分鐘。等候 30 分鐘，然後再次檢查叢集的狀態。

叢集的狀態從 CREATING 到 FAILED

請嘗試再次建立叢集。

叢集狀態為 ACTIVE，但生產者無法傳送資料或取用者無法接收資料

- 如果叢集建立成功 (叢集狀態為 ACTIVE)，但您無法傳送或接收資料，請確定您的生產者和取用者應用程式可以存取叢集。如需詳細資訊，請參閱 the section called “步驟 3：建立用戶端機器” 中的指導。

- 如果您的生產者和取用者可以存取叢集，但仍然遇到產生和使用數據的問題，原因可能是 KAFKA-7697，這會影響 Apache 2.1.0 版本，並可能導致一或多個代理程式死鎖。請考慮遷移到 Apache 卡夫卡 2.2.1，其不受此錯誤影響。如需有關如何遷移的資訊，請參閱 遷移。

AWS CLI 無法識別 Amazon MSK

如果您已 AWS CLI 安裝，但無法識別 Amazon MSK 命令，請升級 AWS CLI 到最新版本。如需如何升級的詳細指示 AWS CLI，請參閱安裝 AWS Command Line Interface. 如需如何使用執行 Amazon MSK 命令的相關資訊，請參閱運作方式。 AWS CLI

分割區離線或複本不同步

這些可能是磁碟空間不足的徵狀。請參閱the section called “磁碟空間不足”。

磁碟空間不足

請參閱管理磁碟空間中的下列最佳實務：the section called “監控磁碟空間” 和 the section called “調整資料保留參數”。

記憶體不足

如果您看到 MemoryUsed 指標過高或 MemoryFree 不足，這並不表示有問題。Apache Kafka 設計為盡可能地多使用記憶體，並且以最佳方式管理。

製片人獲取 NotLeaderForPartitionException

這通常是暫時性錯誤。將生產者的 retries 組態參數設定為高於目前的值。

複寫不足道分區 (URP) 數量大於零

UnderReplicatedPartitions 是重要的監控指標。在狀態良好的 MSK 叢集中，此指標的值為 0。如果它大於零，可能是由於下列原因之一。

- 如果 UnderReplicatedPartitions 是尖峰，問題可能是叢集未以適當的大小佈建，以處理傳入和傳出的流量。請參閱最佳實務。
- 如果 UnderReplicatedPartitions 一直大於 0 (包括在低流量期間)，問題可能是您設定限制性 ACL，不授予主題對代理程式的存取權。若要複寫分割區，必須授權代理程式 READ 和 DESCRIBE 主題。READ 授權預設會授與 DESCRIBE 權限。如需有關設定 ACL 的詳細資訊，請參閱 Apache Kafka 文件中的 授權和 ACL。

叢集具有名為 __amazon_msk_canary 和 __amazon_msk_canary_state 的主題

您可能會看到 MSK 叢集具有名為 __amazon_msk_canary 的主題，以及另一個名為 __amazon_msk_canary_state 的主題。這些是 Amazon MSK 針對叢集運作狀態和診斷指標建立和使用的內部主題。這些主題的大小可以忽略不計，且無法被刪除。

分區複寫失敗

請確定您尚未在 CLUSTER_ACTIONS 上設定 ACL。

無法存取已開啟公開存取的叢集

如果您的叢集已開啟公開存取，但仍無法從網際網路存取，請依照下列步驟執行：

1. 確定叢集安全群組的傳入規則允許您的 IP 地址和叢集的連接埠。如需叢集連接埠號碼的清單，請參閱the section called “連接埠資訊”。同時確保安全群組的傳出規則允許傳出通訊。如需有關安全群組與其傳入、傳出規則的詳細資訊，請參閱《Amazon VPC 使用者指南》中的 VPC 的安全群組。
2. 確定叢集 VPC 網路 ACL 的傳入規則中，允許您的 IP 地址和叢集的連接埠。與安全群組不同，網路 ACL 是無狀態的。這意味著您必須同時設定傳入和傳出規則。在傳出規則中，允許所有流量 (連接埠範圍：0-65535) 傳送到您的 IP 地址。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的新增和刪除規則。
3. 確定您是使用公開存取 bootstrap-broker 字串來存取叢集。開啟公開存取的 MSK 叢集有兩個不同的引導代理程式字串，一個用於公開存取，另一個用於從 AWS內部存取。如需詳細資訊，請參閱the section called “使用獲取引導代理 AWS Management Console”。

無法從內部存取叢集 AWS：網路問題

如果您有一個 Apache Kafka 應用程式無法與 MSK 叢集成功通訊，首先執行以下連線測試。

1. 使用 the section called “取得引導代理程式” 中描述的任何方法來取得引導代理程式的地址。
2. 在下面的命令中，使用您在上一步中獲得的代理程式地址之一取代 *bootstrap-broker*。如果叢集設定為使用 TLS 驗證，請將 *port-number* 替換為 9094。如果叢集不使用 TLS 驗證，請將 *port-number* 替換為 9092。從用戶端機器執行命令。

   ```
   telnet bootstrap-broker port-number
   ```

   其中連接埠號碼為：

   - 如果叢集設定為使用 TLS 驗證，則為 9094。
   - 9092 如果叢集不使用 TLS 驗證。
   - 如果啟用了公用存取，則需要不同的連接埠號碼。

   從用戶端機器執行命令。
3. 對所有引導代理程式重複前面的命令。

如果用戶端機器能夠存取代理程式，則表示沒有連線問題。在這種情況下，執行下列命令來檢查您的 Apache Kafka 用戶端是否已正確設定。若要取得 *bootstrap-brokers*，請使用 the section called "取得引導代理程式" 中描述的任何方法。將 *##* 替換為您的主題名稱。

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

如果上一個命令成功，這表示您的用戶端已正確設定。如果您仍然無法從應用程式生產和使用，請在應用程式層級偵錯問題。

如果用戶端機器無法存取代理程式，請參閱下列小節，以取得根據用戶端機器設定的指引。

同一個 VPC 中的 Amazon EC2 用戶端和 MSK 叢集

如果用戶端機器與 MSK 叢集在相同的 VPC 中，請確定叢集的安全群組具有的傳入規則可接受來自用戶端機器安全群組的流量。如需設定這些規則的資訊，請參閱安全群組規則。如需如何從位於與叢集相同 VPC 中 Amazon EC2 執行個體存取叢集的範例，請參閱開始使用。

不同 VPC 中的 Amazon EC2 用戶端和 MSK 叢集

如果用戶端機器和叢集位於兩個不同的 VPC 中，請確定下列各項：

- 兩個 VPC 是對等的。
- 對等連線的狀態為作用中。
- 兩個 VPC 的路由表已正確設定。

如需 VPC 對等連線的相關資訊，請參閱 使用 VPC 對等連線。

內部部署用戶端

如果是設定為使用連線至 MSK 叢集的內部部署用戶端 AWS VPN，請確定下列事項：

- VPN 連線狀態為 UP。如需如何檢查 VPN 連線狀態的詳細資訊，請參閱如何檢查 VPN 通道的目前狀態？。
- 叢集 VPC 的路由表包含目標具有格式 `Virtual private gateway(vgw-xxxxxxxx)` 的內部部署 CIDR 路由。
- MSK 叢集的安全群組允許連接埠 2181、連接埠 9092 (如果您的叢集接受純文字流量) 和連接埠 9094 (如果您的叢集接受 TLS 加密的流量) 上的流量。

如需詳細 AWS VPN 疑難排解指引，請參閱疑難排解 Client VPN

AWS Direct Connect

如果用戶端使用 AWS Direct Connect，請參閱疑難排解 AWS Direct Connect。

如果先前的故障診斷指南無法解決問題，請確定沒有防火牆封鎖網路流量。若要進一步偵錯，請使用類似 `tcpdump` 和 `Wireshark` 的工具來分析流量，並確定流量已到達 MSK 叢集。

身分驗證失敗：太多連線

`Failed authentication ... Too many connects` 錯誤表示代理程式正在自我保護，因為一個或多個 IAM 用戶端太積極嘗試建立連線。若要讓代理程式接受更高的新 IAM 連線速率，您可以增加 `reconnect.backoff.ms` 組態參數值。

若要深入了解每個代理程式新連線的速率限制，請參閱Amazon MSK 配額頁面。

MSK Serverless：叢集建立失敗

如果您嘗試建立 MSK Serverless 叢集，但工作流程失敗，則您可能沒有建立 VPC 端點的許可。允許 `ec2:CreateVpcEndpoint` 動作，確認您的管理員已授予建立 VPC 端點的許可。

如需有關執行所有 Amazon MSK 動作所需許可的完整清單，請參閱 AWS 管理策略：亞馬遜 FullAccess。

最佳實務

本主題概述一些使用 Amazon MSK 時應遵循的最佳實務。

適當調整叢集大小：每個代理程式的分區數量

下表顯示每個代理程式建議的分區數量上限 (包括領導者和追隨複本)。

經紀人規模	每個代理程式的建議分區數量 (包括領導者和追隨複本)
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

如果每個代理程式的分區數量超過建議值，且叢集超載，您可能無法執行下列操作：

- 更新叢集的組態
- 將叢集更新為較小的代理程式大小
- 將 AWS Secrets Manager 密碼與具有 SASL/SCRAM 驗證的叢集建立關聯

大量的分區也可能導致在 Prometheus 刮擦上 CloudWatch 和丟失卡夫卡指標。

如需選擇分割區數目的指導，請參閱 Apache Kafka 支援每個叢集 200K 個分割區。我們還建議您執行自己的測試，以確定適合您的經紀人的規模。有關不同經紀人大小的更多信息，請參閱the section called “經紀人規模”。

適當調整叢集大小：每個叢集的代理程式數量

若要決定適合您 MSK 叢集的代理程式數量並了解費用，請參閱 MSK 大小與定價試算表。此試算表與類似的、自我管理、以 EC2 為基礎的 Apache Kafka 叢集比較，提供 MSK 叢集大小的預估，以及與 Amazon MSK 相關的費用。若要取得有關試算表中輸入參數的更多資訊，請將游標停留在參數描述上。此表提供的估計值較保守，僅為新叢集提供起點。叢集效能、大小和成本取決於您的使用案例，建議您透過實際測試進行驗證。

若要瞭解基礎結構如何影響 Apache Kafka 效能，請參閱大數據部落格中最佳化 Apache Kafka 叢集大小以最佳化效能和成本的最佳做法。 AWS 部落格文章提供如何調整叢集大小以符合輸送量、可用性和延遲需求的相關資訊。它也提供對一些問題的解答，例如應該何時縱向擴展或橫向擴展，以及如何持續驗證生產叢集大小。

最佳化 m5.4xl、m7g.4xl 或更大型執行個體的叢集輸送量

使用 m5.4xl、m7g.4xl 或更大的執行個體時，您可以透過調整 num.io.thread 和 num.network.thread 組態來最佳化叢集輸送量。

Num.io.Threads 是代理程式用於處理請求的執行緒數量。新增更多執行緒 (最多可支援執行個體大小的 CPU 核心數)，有助於提升叢集輸送量。

Num.network.Threads 是代理程式用於接收所有傳入請求和傳回回應的執行緒數量。網路執行緒將傳入請求放置在請求隊列中，以便由 io.threads 處理。將 num.network.threads 設定為執行個體大小支援的 CPU 核心數量的一半，可完全使用新的執行個體大小。

> **Important**
>
> 在沒有先增加 num.io.threads 的情況下，請勿增加 num.network.threads，因為這可能會導致與佇列飽和度相關的堵塞。

建議設定

執行個體大小	num.io.threads 的建議值	num.network.threads 的建議值
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

使用最新的卡夫卡 AdminClient 以避免主題 ID 不匹配問題

當您使用帶有旗標`--zookeeper`的 Kafka AdminClient 版本低於 2.8.0，以增加或重新指派使用 Kafka 2.8.0 版或更新版本的叢集主題分割區時，會遺失主題識別碼 (錯誤：不符合分割區的主題識別碼)。請注意，`--zookeeper` 旗標已在 Kafka 2.5 棄用，並從 Kafka 3.0 開始被刪除。請參閱 Upgrading to 2.5.0 from any version 0.8.x through 2.4.x。

為了防止主題 ID 不相符，請使用 Kafka 用戶端 2.8.0 或更高版本進行 Kafka 管理員操作。2.5 及更高版本的用戶端可以使用 `--bootstrap-servers` 旗標而不是 `--zookeeper` 旗標。

建置高可用性叢集

請使用下列建議，讓 MSK 叢集在更新期間 (例如，當您更新代理程式大小或 Apache Kafka 版本時) 或 Amazon MSK 正在取代代理程式時具有高可用性。

- 設定一個三可用區域叢集。
- 確保複寫係數 (RF) 至少為 3。請注意，RF 為 1 可能會導致滾動式更新期間分區離線；而 RF 2 可能會導致資料遺失。

- 請將最小同步複本 (minISR) 設定為不超過 RF-1。等於 RF 的 minISR 可避免在滾動更新期間產生至叢集。minISR 2 允許當一個複本離線時，可以使用三向複寫的主題。
- 確定用戶端連線字串至少包含每個可用區域中一個代理程式。如果用戶端連接字串中有多個代理程式，則允許在特定代理程式離線進行更新時進行容錯移轉。如需有關如何取得具有多個代理程式的連接字串的詳細資訊，請參閱the section called “取得引導代理程式”。

監控 CPU 用量

Amazon MSK 強烈建議您將代理程式的總 CPU 使用率 (定義為 `CPU User + CPU System`) 維持在 60% 以下。如果至少有叢集總 CPU 的 40% 可用，Apache Kafka 就能在必要時於叢集中的代理程式之間重新分配 CPU 負載。這種情況的範例之一就是 Amazon MSK 偵測到代理程式故障並進行復原時；在此情況下，Amazon MSK 會執行自動維護，例如修補。另一個範例是使用者請求代理程式大小變更或版本升級時；在這兩種情況下，Amazon MSK 會部署滾動式工作流程，一次讓一個代理程式離線。當具有領導者分區的代理程式離線時，Apache Kafka 會重新分配分區領導者，以將工作重新分配給叢集中的其他代理程式。遵循此最佳實務就能確保叢集中有足夠的 CPU 預留空間來處理這類作業事件。

您可以使用 Amazon CloudWatch 指標數學來建立複合指標`CPU User + CPU System`。設定當複合指標達到平均 60% 的 CPU 使用率時觸發警示。觸發此警示後，請使用下列選項之一擴展叢集：

- 選項 1（建議）：將代理程式大小更新為下一個較大的大小。例如，如果目前的大小為`kafka.m5.large`，請更新要使用的叢集`kafka.m5.xlarge`。請記住，當您更新叢集中的代理程式大小時，Amazon MSK 會以滾動的方式將代理程式離線，並暫時將分區領導地位重新指派給其他經紀人。更新每個代理程式的大小通常需要 10-15 分鐘。
- 選項 2：如果有主題包含從使用循環配置寫入的生產者擷取的所有訊息 (換句話說，訊息不用鍵入且排序對取用者不重要)，請新增代理程式以擴充叢集。同時新增分區至有具有最高輸送量的現有主題。接下來，使用 `kafka-topics.sh --describe` 以確保新增的分區被指派給新的代理程式。與前一個選項相比，此選項主要的好處是您可以更精細地管理資源和成本。此外，如果 CPU 負載大幅超過 60%，也適合使用此選項，因為這種形式的擴展通常不會增加現有代理程式的負載。
- 選項 3：新增代理程式以擴充叢集，然後使用名為 `kafka-reassign-partitions.sh` 的分區重新指派工具來重新指派現有的分區。但是，如果您使用此選項，則重新指派分區之後，叢集將需要花費資源將資料從一個代理程式複製到另一個。與之前的兩個選項相比，這個選項在最初會顯著增加叢集的負載。因此，當 CPU 使用率超過 70% 時，Amazon MSK 不建議使用此選項，因為複製會造成額外的 CPU 負載和網路流量。只有在先前兩個選項都不可行時，Amazon MSK 才建議使用此選項。

其他建議：

- 監控每個代理程式的總 CPU 使用率，將其作為負載分配的測算代替物。如果代理程式的 CPU 使用率持續不均，則可能代表負載未均勻分佈在叢集中。Amazon MSK 建議使用 Cruise Control，透過指派分區持續管理負載分配。
- 監控生產和取用延遲。生產和取用延遲會隨著 CPU 使用率線性增加。
- JMX 湊集間隔：如果您使用 Prometheus 功能啟用了開放式監控，建議您對 Prometheus 主機組態 (prometheus.yml) 使用 60 秒或更高的湊集間隔 (scrape_interval：60s)。降低湊集間隔可能導致叢集上的 CPU 使用率過高。

監控磁碟空間

若要避免訊息的磁碟空間不足，請建立監視`KafkaDataLogsDiskUsed`指標的 CloudWatch 警示。當此指標的值達到或超過 85% 時，請執行下列一或多個動作：

- 請使用 the section called “自動調整規模”。您也可以手動增加代理程式儲存空間，如 the section called “手動擴展” 中所述。
- 縮短郵件保留期間或記錄大小。如需如何執行此作業的資訊，請參閱 the section called “調整資料保留參數”。
- 刪除未使用的主題。

如需如何設定和使用警示的詳細資訊，請參閱使用 Amazon CloudWatch 警示。如需 Amazon MQ 指標的完整清單，請參閱監控叢集。

調整資料保留參數

取用訊息不會從日誌中刪除它們。若要定期釋放磁碟空間，您可以明確指定保留期間，也就是訊息在記錄中保留的時間長度。您也可以指定保留記錄大小。當無論是保留時間週期或保留記錄大小達到，Apache Kafka 會開始從記錄中刪除非作用中區段。

若要在叢集層級指定保留政策，請設定下列一或多個參數：`log.retention.hours`、`log.retention.minutes`、`log.retention.ms` 或 `log.retention.bytes`。如需詳細資訊，請參閱 the section called “自訂組態”。

您也可以在主題層級指定保留參數：

- 若要指定每個主題保留時間期間，請使用下列命令。

```
kafka-configs.sh --bootstrap-server $bs --alter --entity-type topics --entity-
name TopicName --add-config retention.ms=DesiredRetentionTimePeriod
```

- 若要指定每個主題的保留記錄大小，請使用下列命令。

```
kafka-configs.sh --bootstrap-server $bs --alter --entity-type topics --entity-
name TopicName --add-config retention.bytes=DesiredRetentionLogSize
```

您在主題層級指定的保留參數會優先於叢集層級參數。

不正常關機後加快復原日誌速度

不正常關機後，代理程式可能需要一段時間才能重新啟動，因為它會試圖復原日誌。根據預設，Kafka 對每個日誌目錄僅使用單執行緒進行復原。例如，如果您有數千個分區，日誌復原可能需要數小時才能完成。若要加速日誌復原，建議使用組態屬性 num.recovery.threads.per.data.dir 增加執行緒數量。您可以將其設定為 CPU 核心數量。

監控 Apache Kafka 記憶體

我們建議您監控 Apache Kafka 的記憶體用量。否則，叢集可能無法使用。

若要判斷 Apache Kafka 使用了多少記憶體，您可以監控 HeapMemoryAfterGC 指標。HeapMemoryAfterGC 則是垃圾回收之後使用中的總堆積記憶體百分比。我們建議您建立 CloudWatch 警報，在HeapMemoryAfterGC增加到 60% 以上時採取行動。

您可以採取以減少記憶體用量的步驟各有不同。它們取決於您設定 Apache Kafka 的方式。例如，如果您使用交易性訊息傳輸，則可以將 Apache Kafka 組態中的 transactional.id.expiration.ms 值從 604800000 毫秒降至 86400000 毫秒 (從 7 天減少為 1 天)。這會減少每個交易的記憶體用量。

不要新增非 MSK 代理程式

對於 ZooKeeper基於叢集，如果您使用 Apache ZooKeeper 命令來新增代理程式，這些代理程式不會新增至您的 MSK 叢集，而且您的 Apache ZooKeeper 將包含有關叢集的錯誤資訊。這可能會導致資料遺失。如需支援的叢集操作，請參閱 運作方式。

啟用傳輸中加密

如需傳輸中加密及如何啟用的相關資訊，請參閱 the section called “傳輸中加密”。

重新指派分割區

若要將分割區移至相同叢集上的不同代理程式，您可以使用名為 `kafka-reassign-partitions.sh` 的分割區重新指派工具。例如，在新增代理程式以擴充叢集或移動分割區以移除代理程式之後，您可以透過將分割區重新指派給新的代理程式來重新平衡該叢集。如需如何新增代理程式至叢集的詳細資訊，請參閱 the section called “擴充叢集”。如需如何從叢集中移除代理程式的相關資訊，請參閱the section called “刪除經紀人”。如需有關分割區重新指派工具的詳細資訊，請參閱 Apache Kafka 文件中的 擴充您的叢集。

《Amazon MSK 開發人員指南》文件歷史記錄

下表說明《Amazon MSK 開發人員指南》的重要變更。

最新文件更新：2024 年 6 月 25 日

變更	描述	日期
添加了引力彈升級功能。	您可以將叢集代理程式的大小從 M5 或 T3 更新為 M7g，或從 M7g 更新為 M5。	2024-6-25
3.4.0 支援結束日期宣布。	阿帕奇卡夫卡版本 3.4.0 的支援日期結束為 2025 年 6 月 17 日。	2024-6-24
添加了代理刪除功能。	您可以透過移除一組代理程式來減少佈建叢集的儲存和運算容量，而不會影響可用性、資料耐久性風險或資料串流應用程式中斷。	2024-5-16
`WriteDataIdempotently` 已新增至 AWSMSKReplicatorExecutionRole	WriteDataIdempotently 權限已新增至 AWSMSKReplicatorExecutionRole 原則，以支援 MSK 叢集之間的資料複寫。	2024-5-16
引力子 M7g 經紀人在巴西和巴林發布。	Amazon MSK 現在支援使用 AWS Graviton 處理器 (由亞馬 Amazon Web Services 務建置的自訂 ARM 處理器) 的 M7g 代理程式提供南美 (sa-east-1、聖保羅) 和中東 (me-south-1、巴林) 區域的可用性。	2024-01-11
向中國地區釋放引力子 M7g 經紀商	Amazon MSK 現在支援使用 AWS 重力處理器 (由亞馬 Amazon Web Services 建立的	2024-01-11

變更	描述	日期
	自訂 ARM 處理器) 的 M7g 代理程式的中國區域可用性。	
Amazon MSK 卡夫卡版支持政策	已新增 Amazon MSK 支援的卡夫卡版本支援政策的說明。如需詳細資訊，請參閱 Apache 卡夫卡版本。	2023-12-08
支援 Amazon MSK 複寫器的新服務執行角色政策。	Amazon MSK 增加了新`AWSMSKReplicatorExecutionRole` 政策以支持 Amazon MSK 複製器。如需詳細資訊，請參閱AWS 受管政策：`AWSMSKReplicatorExecutionRole` 。	2023-12-06
M7 克重力彈支撐	Amazon MSK 現在支援使用 AWS 重力發處理器的 M7g 代理程式 (由 Amazon Web Services 建置的自訂 ARM 處理器)。	2023-11-27
Amazon MSK Replicator	Amazon MSK Replicator 是一項新功能，可用來在 Amazon MSK 叢集之間複寫資料。Amazon MSK 複製器包含亞馬遜 FullAccess MSK 政策的更新。如需詳細資訊，請參閱AWS 受管政策：`AmazonMSKFullAccess` 。	2023-09-28
已更新 IAM 最佳實務。	更新了指南以符合 IAM 最佳實務。如需更多詳細資訊，請參閱 IAM 中的安全最佳實務。	2023-03-08

變更	描述	日期
服務連結角色更新以支援多 VPC 私有連線	Amazon MSK 現在包含 AWSServiceRoleForKafka 服務連結角色更新，可管理帳戶中的網路界面和 VPC 端點，讓叢集代理程式可供 VPC 中的用戶端存取。Amazon MSK 會使用 `DescribeVpcEndpoints` 、`ModifyVpcEndpoint` 和 `DeleteVpcEndpoints` 許可。如需詳細資訊，請參閱使用 Amazon MSK 的服務連結角色。	2023-03-08
支援 Apache Kafka 2.7.2	Amazon MSK 現已支援 Apache Kafka 2.7.2 版。如需詳細資訊，請參閱 支援的 Apache Kafka 版本。	2021-12-21
支援 Apache Kafka 2.6.3	Amazon MSK 現已支援 Apache Kafka 2.6.3 版。如需詳細資訊，請參閱 支援的 Apache Kafka 版本。	2021-12-21
MSK Serverless 發行前版本	MSK Serverless 是可用來建立無伺服器叢集的新功能。如需詳細資訊，請參閱 MSK Serverless。	2021-11-29
支援 Apache Kafka 2.8.1	Amazon MSK 現已支援 Apache Kafka 2.8.1 版。如需詳細資訊，請參閱 支援的 Apache Kafka 版本。	2021-09-30

變更	描述	日期
MSK Connect	MSK Connect 是一項新功能，可以使用其來建立和管理 Apache Kafka 連接器。如需詳細資訊，請參閱 MSK Connect。	2021-09-16
支援 Apache Kafka 2.7.1	Amazon MSK 現已支援 Apache Kafka 2.7.1 版。如需詳細資訊，請參閱 支援的 Apache Kafka 版本。	2021-05-25
支援 Apache Kafka 2.8.0	Amazon MSK 現已支援 Apache Kafka 2.8.0 版。如需詳細資訊，請參閱 支援的 Apache Kafka 版本。	2021-04-28
支援 Apache Kafka 2.6.2	Amazon MSK 現已支援 Apache Kafka 2.6.2 版。如需詳細資訊，請參閱 支援的 Apache Kafka 版本。	2021-04-28
支援更新代理程式類型	您現在可以變更現有叢集的代理程式類型。如需詳細資訊，請參閱 更新代理大小。	2021-01-21
支援 Apache Kafka 2.6.1	Amazon MSK 現已支援 Apache Kafka 2.6.1 版。如需詳細資訊，請參閱 支援的 Apache Kafka 版本。	2021-01-19
支援 Apache Kafka 2.7.0	Amazon MSK 現已支援 Apache Kafka 2.7.0 版。如需詳細資訊，請參閱 支援的 Apache Kafka 版本。	2020-12-29

變更	描述	日期
不再有使用 Apache Kafka 1.1.1 版的新叢集	您無法再使用 Apache Kafka 1.1.1 版來建立新的 Amazon MSK 叢集。然而，若您有正在執行 Apache Kafka 1.1.1 版的現有 MSK 叢集，可以繼續在這些現有叢集上使用目前支援的所有功能。如需詳細資訊，請參閱 Apache Kafka 版本。	2020-11-24
取用者延遲指標	Amazon MSK 現在會提供指標，供您用來監控取用者延遲。如需詳細資訊，請參閱 監控 Amazon MSK 叢集。	2020-11-23
支援 Cruise Control	Amazon MSK 現在支持LinkedIn的巡航控制。如需詳細資訊，請參閱 使用 LinkedIn 的巡航控制阿帕奇卡夫卡與 Amazon MSK。	2020-11-17
支援 Apache Kafka 2.6.0	Amazon MSK 現已支援 Apache Kafka 2.6.0 版。如需詳細資訊，請參閱 支援的 Apache Kafka 版本。	2020-10-21
支援 Apache Kafka 2.5.1	Amazon MSK 現已支援 Apache Kafka 2.5.1 版。透過 Apache 卡夫卡 2.5.1 版，Amazon MSK 支援用戶端和端點之間傳輸過程中的加密。 ZooKeeper 如需詳細資訊，請參閱 支援的 Apache Kafka 版本。	2020-09-30

變更	描述	日期
應用程式自動擴展	您可以設定 Amazon Managed Streaming for Apache Kafka，以自動擴展叢集的儲存來因應用量增加的情況。如需詳細資訊，請參閱 自動調整規模。	2020-09-30
支援使用者名稱和密碼安全	Amazon MSK 現已支援利用使用者名稱和密碼登入叢集。Amazon MSK 在 AWS Secrets Manager 中存儲登入資料。如需詳細資訊，請參閱 SASL/SCRAM 身分驗證。	2020-09-17
支援升級 Amazon MSK 叢集的 Apache Kafka 版本	您現在可以更新現有 MSK 叢集的 Apache Kafka 版本。	2020-05-28
支援 T3.small 中介裝置節點	Amazon MSK 現已支援建立具有 Amazon EC2 T3.small 類型的代理程式的叢集。	2020 年 4 月 8 日
Apache Kafka 2.4.1 的支援	Amazon MSK 現已支援 Apache Kafka 2.4.1 版。	2020-04-02
支援串流代理程式日誌	Amazon MSK 現在可以將代理程式日誌串流到 CloudWatch 日誌、Amazon S3 和 Amazon 資料 Firehose。Firehose 可以反過來將這些記錄傳送至其支援的目的地，例如「 OpenSearch 服務」。	2020-02-25
Apache Kafka 2.3.1 的支援	Amazon MSK 現已支援 Apache Kafka 2.3.1 版。	2019-12-19
開放式監控	Amazon MSK 現已支援使用 Prometheus 進行開放式監控。	2019-12-04

變更	描述	日期
Apache Kafka 2.2.1 的支援	Amazon MSK 現已支援 Apache Kafka 2.2.1 版。	2019-07-31
一般可用性	新功能包含標記支援、身分驗證、TLS 加密、設定，以及更新代理程式儲存體的能力。	2019-05-30
Apache Kafka 2.1.0 的支援	Amazon MSK 現已支援 Apache Kafka 2.1.0 版。	2019-02-05

AWS 詞彙表

有關最新 AWS 術語，請參閱AWS 詞彙表 參考文獻中的AWS 詞彙表。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。