aws

使用者指南

AWS Elemental MediaStore

AWS Elemental MediaStore: 使用者指南

Table of Contents

支援結束通知：2025 年 11 月 13 日， AWS 將停止對 AWS Elemental 的支援 MediaStore。2025 年 11 月 13 日之後，您將無法再存取 MediaStore 主控台或 MediaStore 資源。如需詳細資訊，請造訪此部落格文章。

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。

什麼是 AWS Elemental MediaStore？

AWS Elemental MediaStore 是一種影片原始和儲存服務，可提供即時原始伺服器所需的高效能和即時一致性。使用 MediaStore，您可以將影片資產管理為容器中的物件，以建置可靠的雲端媒體工作流程。

若要使用 服務，您可以將物件從編碼器或資料饋送等來源上傳至您在 中建立的容器 MediaStore。

MediaStore 當您需要強大的一致性、低延遲的讀取和寫入，以及能夠處理大量並行請求時， 是儲存分段影片檔案的絕佳選擇。如果您未提供即時串流影片，請考慮改用 Amazon Simple Storage Service（Amazon S3）。

主題

AWS 元素 MediaStore 概念和術語

ARN

Amazon Resource Name。

Body

要上傳至物件中的資料。

(位元組) 範圍

要加以定址之物件資料的子集。如需詳細資訊，請參閱 HTTP規格的範圍。

容器

保存物件的命名空間。容器會有一個您可用於附加寫入和擷取物件以及附加存取政策的端點。

端點

MediaStore 服務的項目點，以HTTPS根 形式提供URL。

ETag

實體標籤，這是物件資料的雜湊值。

資料夾

容器的劃分區。資料夾可以保存物件及其他資料夾。

項目

用來指稱物件及資料夾的詞彙。

物件

類似 Amazon S3 物件 的資產。物件是存放在 中的基本實體 MediaStore。此服務接受所有檔案類型。

發送服務

MediaStore 被視為原始服務，因為它是媒體內容交付的分發點。

路徑

物件或資料夾的唯一識別符，這可表示其在容器中的位置。

部分

物件的資料子集 (區塊)。

政策

IAM 政策 。

資源

您可以在 中使用 AWS 的實體。每個AWS資源都會指派一個 Amazon Resource Name （ ARN ），做為唯一的識別符。在 中 MediaStore，這是資源及其ARN格式：

- 容器：`aws:mediastore:region:account-id:container/:containerName`

相關服務

- Amazon CloudFront 是一項全球內容交付網路 （ CDN ） 服務，可安全地將資料和影片交付給您的觀眾。使用 CloudFront 以最佳效能交付內容。如需詳細資訊，請參閱 Amazon CloudFront 開發人員指南 。

- AWS CloudFormation 是一項協助您建立模型和設定 AWS 資源的服務。您可以建立範本來描述您想要的所有 AWS 資源（例如 MediaStore 容器），並 AWS CloudFormation 負責為您佈建和設定這些資源。您不需要個別建立和設定 AWS 資源，並了解什麼取決於什麼； AWS CloudFormation 處理所有這些。如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/。
- AWS CloudTrail 是一項服務，可讓您監控對 帳戶的呼叫 CloudTrail API，包括AWS由 管理主控台進行的呼叫 AWS CLI，以及其他 服務。如需詳細資訊，請參閱《AWS CloudTrail 使用者指南》https://docs.aws.amazon.com/awscloudtrail/latest/userguide/。
- Amazon CloudWatch 是 AWS Cloud 資源和您在 上執行之應用程式的監控服務 AWS。使用 CloudWatch 事件來追蹤 中容器和物件狀態的變更 MediaStore。如需詳細資訊，請參閱 Amazon CloudWatch 文件 。
- AWS Identity and Access Management（IAM）是一種 Web 服務，可協助您安全地控制使用者對 AWS 資源的存取。使用 IAM 控制誰可以使用您的 AWS 資源（身分驗證），以及使用者可以使用哪些資源（授權）。如需詳細資訊，請參閱設定AWS元素 MediaStore。
- Amazon Simple Storage Service（Amazon S3）是物件儲存體，用於從任何地方存放和擷取任何數量的資料。如需詳細資訊，請參閱 Amazon S3 說明文件。

存取 AWS Elemental MediaStore

您可以使用下列 MediaStore 任一方法存取 ：

- AWS 管理主控台 – 本指南中的程序說明如何使用AWS管理主控台來執行 的任務 MediaStore。若要 MediaStore 使用主控台存取 ：

```
https://<region>.console.aws.amazon.com/mediastore/home
```

- AWS Command Line Interface – 如需詳細資訊，請參閱 AWS Command Line Interface 使用者指南 。若要 MediaStore 使用CLI端點存取 ：

```
aws mediastore
```

- MediaStore API – 如果您使用的是 SDK 無法使用的程式設計語言，請參閱 AWS Elemental MediaStore API 參考，以取得有關API動作以及如何提出API請求的資訊。若要 MediaStore 使用 RESTAPI端點存取 ：

```
https://mediastore.<region>.amazonaws.com
```

- AWS SDKs – 如果您使用的程式設計語言為 AWS 提供 SDK，您可以使用 SDK來存取 MediaStore。SDKs 簡化身分驗證、輕鬆與您的開發環境整合，以及輕鬆存取 MediaStore 命令。如需詳細資訊，請參閱 Amazon Web Services 適用工具。
- AWS 適用於 Windows 的工具 PowerShell – 如需詳細資訊，請參閱 AWS Tools for Windows PowerShell 使用者指南 。

AWS Elemental 的定價 MediaStore

與其他 AWS 產品一樣，使用 沒有合約或最低承諾 MediaStore。我們向您收取的費用僅限內容進入服務時的每 GB 提取費，以及您在服務中存放內容的每 GB 月費。如需詳細資訊，請參閱 AWS Elemental MediaStore Pricing 。

AWS Elemental 的區域和端點 MediaStore

若要減少應用程式中的資料延遲， MediaStore 會提供區域端點來提出請求：

```
https://mediastore.<region>.amazonaws.com
```

若要檢視 MediaStore 可用AWS區域的完整清單，請參閱 AWS一般參考中的AWS元素 MediaStore 端點和配額。

設定AWS元素 MediaStore

本節將引導您完成設定使用者存取 AWS Elemental 所需的步驟 MediaStore。如需有關的身分識別與存取管理的背景和其他資訊 MediaStore，請參閱AWS Elemental 的 Identity and Access Management MediaStore。

要開始使用AWS元素 MediaStore，請完成以下步驟。

主題

- 註冊一個 AWS 帳戶
- 建立具有管理存取權的使用者

註冊一個 AWS 帳戶

如果您沒有 AWS 帳戶，請完成下列步驟來建立一個步驟。

若要註冊成為 AWS 帳戶

1. 打開https://portal.aws.amazon.com/billing/註冊。
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 AWS 帳戶，一個 AWS 帳戶根使用者已建立。根使用者可以存取所有 AWS 服務 和帳戶中的資源。作為安全最佳實務，請將管理存取權指派給使用者，並且僅使用根使用者來執行需要根使用者存取權的任務。

AWS 註冊過程完成後，會向您發送確認電子郵件。您可以隨時前往 https://aws.amazon.com/並選擇「我的帳戶」，檢視目前的帳戶活動並管理您的帳戶。

建立具有管理存取權的使用者

在您註冊一個 AWS 帳戶，保護您的 AWS 帳戶根使用者，啟用 AWS IAM Identity Center，並建立系統管理使用者，這樣您就不會將 root 使用者用於日常工作。

保護您的 AWS 帳戶根使用者

1. 登入 AWS Management Console通過選擇 Root 用戶並輸入您的帳戶所有者 AWS 帳戶 電子郵件地址。在下一頁中，輸入您的密碼。

   如需使用 root 使用者登入的說明，請參閱以 root 使用者身分登入 AWS 登入 使用者指南。
2. 為您的 root 使用者開啟多因素驗證 (MFA)。

   如需指示，請參閱為您的MFA裝置啟用虛擬裝置 AWS 帳戶 使用者指南中的 root IAM 使用者 (主控台)。

建立具有管理存取權的使用者

1. 啟用IAM身分識別中心。

   如需指示，請參閱啟用 AWS IAM Identity Center 中的 AWS IAM Identity Center 使用者指南。
2. 在IAM身分識別中心中，將管理存取權授與使用者。

   若要取得有關使用 IAM Identity Center 目錄 做為您的身分識別來源，請參閱以預設值設定使用者存取 IAM Identity Center 目錄 中的 AWS IAM Identity Center 使用者指南。

以具有管理存取權的使用者身分登入

- 若要使用您的 IAM Identity Center 使用者登入URL，請使用建立IAM身分識別中心使用者時傳送至您電子郵件地址的登入資訊。

  如需使用IAM身分識別中心使用者登入的說明，請參閱登入 AWS 存取入口網站 AWS 登入 使用者指南。

指派存取權給其他使用者

1. 在 IAM Identity Center 中，建立遵循套用最低權限權限的最佳作法的權限集。

   如需指示，請參閱 AWS IAM Identity Center 使用者指南。
2. 將使用者指派至群組，然後對該群組指派單一登入存取權。

   如需指示，請參閱新增群組 AWS IAM Identity Center 使用者指南。

AWS Elemental 入門 MediaStore

本入門教學課程說明如何使用 AWS Elemental MediaStore 建立容器和上傳物件。

主題

步驟 1：存取 AWS Elemental MediaStore

設定 AWS 帳戶並建立使用者和角色之後，您就可以登入 AWS Elemental 的主控台 MediaStore。

若要存取 AWS Elemental MediaStore

- 請登入AWS Management Console並開啟 MediaStore 主控台，網址為 https://console.aws.amazon.com/mediastore/。

> **Note**
>
> 您可以使用您已為此帳戶建立的任何 IAM 登入資料來登入。如需有關建立 IAM 登入資料的詳細資訊，請參閱設定AWS元素 MediaStore。

步驟 2：建立容器

您可以使用 AWS Elemental 中的容器 MediaStore 來存放資料夾和物件。您可以使用容器將相關物件分組，其方式與使用目錄將檔案系統中的檔案分組相同。建立容器時，您不需付費；只有在您上傳物件到容器時才付費。

若要建立容器

1. 在 Containers (容器) 頁面上，選擇 Create container (建立容器)。
2. 在 Container name (容器名稱) 中，輸入容器的名稱。如需詳細資訊，請參閱容器名稱規則。

3. 選擇 [建立容器]。AWS Elemental 會 MediaStore 將新容器新增至容器清單中。容器一開始的狀態會是 Creating (建立中)，然後變更為 Active (啟用)。

步驟 3：上傳物件

您可以將物件 (各物件最多 25 MB) 上傳至容器或容器中的資料夾。若要將物件上傳至資料夾，請指定資料夾的路徑。如果資料夾已存在，AWS Elemental 會將物件 MediaStore 儲存在資料夾中。如果資料夾不存在，則服務會建立此資料夾，再將物件存放在其中。

> **Note**
>
> 物件檔案名稱只能包含字母、數字、句點 (.)、底線 (_)、波狀符號 (~) 和連字號 (-)。

若要上傳物件

1. 在 Containers (容器) 頁面上，選擇您剛才建立的容器的名稱。容器的詳細資訊頁面隨即出現。
2. 選擇 Upload object (上傳物件)。
3. 在 Target path (目標路徑) 中，輸入資料夾的路徑。例如：`premium/canada`。如果路徑中的任何資料夾尚未存在，AWS Elemental 會自動 MediaStore 建立這些資料夾。
4. 針對 Object (物件) 選擇 Browse (瀏覽)。
5. 導覽至適當的資料夾，然後選擇一個物件來上傳。
6. 選擇 Open (開啟)，然後選擇 Upload (上傳)。

步驟 4：存取物件

您可以將物件下載到指定的端點。

1. 在 Containers (容器) 頁面上，選擇有您所要下載之物件的容器的名稱。
2. 如果您想要下載的物件是在某個子資料夾中，請繼續選擇資料夾名稱，直到您看到該物件為止。
3. 選擇物件的名稱。
4. 在物件的詳細資訊頁面上，選擇 Download (下載)。

AWS Elemental 中的容器MediaStore

您可以使用 MediaStore 中的容器來存放您的資料夾和物件。相關物件可以在容器中進行分組，其方式與使用目錄將檔案系統中的檔案分組相同。建立容器時，您不需付費；只有在您上傳物件到容器時才付費。如需費用的詳細資訊，請參AWS ElementalMediaStore定價。

主題

容器名稱規則

為容器選擇名稱時，請記得下列項目：

- 此名稱在目前帳戶的目前 AWS 區域中必須是唯一的。
- 此名稱可以包含大寫字母、小寫字母、數字和底線 (_)。
- 名稱長度必須介於 1 至 255 個字元。
- 名稱區分大小寫。例如，您可以有名為 `myContainer` 的容器以及名為 `mycontainer` 的資料夾，因為這些名稱是唯一的。
- 不可在建立容器後加以重新命名。

建立容器

您可以為每個 AWS 帳戶建立最多 100 個容器。您可以視需要建立任何數量的資料夾，只要容器中的巢狀層級不超過 10 層。此外，您可以將任意數量的物件上傳到每個容器。

> **Tip**
>
> 您也可以透過使用 AWS CloudFormation 範本來自動建立容器。AWS CloudFormation 範本可管理五個 API 動作的資料：建立容器、設定存取記錄、更新預設容器政策、新增跨來

源資源共享 (CORS) 政策，以及新增物件生命週期政策。如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》。

若要建立容器 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇 Create container (建立容器)。
3. 在 Container (容器) 名稱中，輸入容器的名稱。如需詳細資訊，請參閱 容器名稱規則。
4. 選擇建立容器。AWS ElementalMediaStore會將新的容器新增至容器清單。容器一開始的狀態會是 Creating (建立中)，然後變更為 Active (啟用)。

若要建立容器 (AWS CLI)

- 在 AWS CLI 中，使用 `create-container` 命令：

```
aws mediastore create-container --container-name ExampleContainer --region us-
west-2
```

以下範例顯示傳回值：

```
{
    "Container": {
        "AccessLoggingEnabled": false,
        "CreationTime": 1563557265.0,
        "Name": "ExampleContainer",
        "Status": "CREATING",
        "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer"
    }
}
```

檢視容器的詳細資訊

容器的詳細資訊包括容器政策、端點、ARN 和建立時間。

若要檢視容器詳細資訊 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇容器的名稱。

   容器詳細資訊頁面隨即出現。此頁面分為兩個區段：

   - Objects (物件) 區段，其中列出容器中的物件和資料夾。
   - Container policy (容器政策) 區段，其中顯示與此容器相關聯之依資源而定的政策。如需資源政策的相關資訊，請參閱容器政策。

若要檢視容器詳細資訊 (AWS CLI)

- 在 AWS CLI 中，使用 `describe-container` 命令：

```
aws mediastore describe-container --container-name ExampleContainer --region us-
west-2
```

以下範例顯示傳回值：

```
{
    "Container": {
        "CreationTime": 1563558086.0,
        "AccessLoggingEnabled": false,
        "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer",
        "Status": "ACTIVE",
        "Name": "ExampleContainer",
        "Endpoint": "https://aaabbbcccdddee.data.mediastore.us-
west-2.amazonaws.com"
    }
}
```

檢視容器清單

您可以檢視所有與您帳戶相關之容器的清單。

若要檢視容器的清單 (主控台)

- 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。

  Containers (容器) 頁面會出現，列出所有與您帳戶相關聯的容器。

若要檢視容器的清單 (AWS CLI)

- 在 AWS CLI 中，使用 `list-containers` 命令。

```
aws mediastore list-containers --region us-west-2
```

以下範例顯示傳回值：

```
{
    "Containers": [
        {
            "CreationTime": 1505317931.0,
            "Endpoint": "https://aaabbbcccdddee.data.mediastore.us-
west-2.amazonaws.com",
            "Status": "ACTIVE",
            "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleLiveDemo",
            "AccessLoggingEnabled": false,
            "Name": "ExampleLiveDemo"
        },
        {
            "CreationTime": 1506528818.0,
            "Endpoint": "https://fffggghhhiiijj.data.mediastore.us-
west-2.amazonaws.com",
            "Status": "ACTIVE",
            "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer",
            "AccessLoggingEnabled": false,
            "Name": "ExampleContainer"
        }
    ]
}
```

刪除容器

只有當容器沒有任何物件時，您才可以將該容器刪除。

若要刪除容器 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇容器名稱左側的選項。
3. 選擇 Delete (刪除)。

若要刪除容器 (AWS CLI)

- 在 AWS CLI 中，使用 `delete-container` 命令：

  ```
  aws mediastore delete-container --container-name=ExampleLiveDemo --region us-west-2
  ```

  此命令沒有傳回值。

AWS Elemental 中的政策MediaStore

您可以將下列一或多個政策套用至 AWS ElementalMediaStore容器：

- 容器政策-設定容器內所有資料夾及物件的存取權利。MediaStore設置默認策略，允許用户執行所有MediaStore操作。此政策指定所有操作必須透過 HTTPS 執行。建立容器之後，您可以編輯容器政策。
- 跨來源資源分享 (CORS) 政策-允許一個網域中的用户端 Web 應用程式與不同網域中的資源互動。MediaStore未設置默認 CORS 策略。
- 指標政策-允許MediaStore將指標傳送至亞馬遜CloudWatch。MediaStore不會設定預設指標政策。
- 物件生命週期政策-控制物件保留在MediaStore容器。MediaStore不會設定預設物件生命週期政策。

AWS 元素中的容器策略MediaStore

每個容器都有依資源而定的政策，可管理該容器中所有資料夾及物件的存取權利。預設政策 (這已自動附加至所有新的容器) 允許存取所有 AWS ElementalMediaStore操作。政策會指定此存取有其必要條件，也就是必須使用 HTTPS 進行操作。建立容器後，您就可以編輯容器中連接的政策。

您也可以指定物件生命週期政策，此政策會在容器中管理物件的過期日期。物件達到您指定的年齡上限後，此服務就會將這些物件從容器中刪除。

主題

檢視容器政策

您可以使用主控台或 AWS CLI 來檢視容器中依資源而定的政策。

若要檢視容器政策 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇容器名稱。

   容器詳細資訊頁面隨即出現。政策會顯示在 Container policy (容器政策) 區段中。

若要檢視容器政策 (AWS CLI)

- 在 AWS CLI 中，使用 `get-container-policy` 命令：

```
aws mediastore get-container-policy --container-name ExampleLiveDemo --region us-
west-2
```

以下範例顯示傳回值：

```
{
  "Policy": {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "PublicReadOverHttps",
        "Effect": "Allow",
        "Principal": {
          "AWS": "arn:aws:iam::111122223333:root",
        },
        "Action": [
          "mediastore:GetObject",
          "mediastore:DescribeObject",
        ],
        "Resource": "arn:aws:mediastore:us-
west-2:111122223333:container/ExampleLiveDemo/*",
        "Condition": {
          "Bool": {
            "aws:SecureTransport": "true"
          }
        }
      }
    ]
  }
}
```

編輯容器政策

您可以編輯預設容器政策中的許可，也可以建立取代預設政策的新政策。新的政策需要最多 5 分鐘就會生效。

若要編輯容器政策 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇容器名稱。
3. 選擇 Edit Policy (編輯政策)。如需說明如何設定不同許可的範例，請參閱the section called “範例容器政策”。
4. 進行適當變更，然後選擇 Save (儲存)。

若要編輯容器政策 (AWS CLI)

1. 建立檔案，此檔案會定義容器政策：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadOverHttps",
      "Effect": "Allow",
      "Action": ["mediastore:GetObject", "mediastore:DescribeObject"],
      "Principal": "*",
      "Resource": "arn:aws:mediastore:us-
west-2:111122223333:container/ExampleLiveDemo/*",
      "Condition": {
        "Bool": {
            "aws:SecureTransport": "true"
        }
      }
    }
  ]
}
```

2. 在 AWS CLI 中，使用 `put-container-policy` 命令：

```
aws mediastore put-container-policy --container-name ExampleLiveDemo --
policy file://ExampleContainerPolicy.json --region us-west-2
```

此命令沒有傳回值。

範例容器政策

下列範例示範針對不同使用者群組所建構的容器政策。

主題

範例容器政策：預設

創建容器時，AWS 元素MediaStore會自動附加下列依資源而定的政策：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MediaStoreFullAccess",
      "Action": [ "mediastore:*" ],
      "Principal":{
          "AWS" : "arn:aws:iam::<aws_account_number>:root"},
      "Effect": "Allow",
      "Resource": "arn:aws:mediastore:<region>:<owner acct number>:container/<container
 name>/*",
      "Condition": {
        "Bool": { "aws:SecureTransport": "true" }
      }
    }
  ]
}
```

政策已內建於服務，因此您不需要建立。不過，您可以編輯政策，如果默認策略中的權限與您想要用於容器的權限不一致，則在容器上。

指派給所有新容器的預設政策允許對容器進行所有 MediaStore 操作存取。政策會指定此存取有其必要條件，也就是必須使用 HTTPS 進行操作。

範例容器政策：透過 HTTPS 的公有讀取存取

此範例政策允許使用者透過 HTTPS 請求擷取物件。此政策允許任何人透過安全的 SSL/TLS 連線進行讀取存取：驗證過身分的使用者以及匿名使用者 (未登入的使用者)。此陳述式的名稱為 PublicReadOverHttps。這允許對任何物件 (如資源路徑結尾的 * 所指定) 進行 GetObject 和 DescribeObject 操作存取。政策會指定此存取有其必要條件，也就是必須使用 HTTPS 進行操作：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadOverHttps",
      "Effect": "Allow",
      "Action": ["mediastore:GetObject", "mediastore:DescribeObject"],
      "Principal": "*",
      "Resource": "arn:aws:mediastore:<region>:<owner acct number>:container/<container
 name>/*",
      "Condition": {
        "Bool": {
            "aws:SecureTransport": "true"
        }
      }
    }
  ]
}
```

範例容器政策：透過 HTTP 或 HTTPS 的公有讀取存取

此範例政策允許對任何物件 (如資源路徑結尾的 * 所指定) 進行 GetObject 和 DescribeObject 操作存取。此政策允許任何人進行讀取存取，包括所有驗證過身分的使用者以及匿名使用者 (未登入的使用者)：

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

```
      "Action": ["mediastore:GetObject", "mediastore:DescribeObject"],
      "Principal": "*",
      "Resource": "arn:aws:mediastore:<region>:<owner acct number>:container/<container
 name>/*",
      "Condition": {
        "Bool": { "aws:SecureTransport": ["true", "false"] }
      }
    }
  ]
}
```

範例容器政策：跨帳戶讀取存取 - 啟用 HTTP

此範例政策允許使用者透過 HTTP 請求擷取物件。這允許具備跨帳戶存取權且驗證過身分的使用者進行此存取。物件不一定要在使用 SSL/TLS 憑證的伺服器上託管：

```
{
  "Version" : "2012-10-17",
  "Statement" : [ {
    "Sid" : "CrossAccountReadOverHttpOrHttps",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::<other acct number>:root"
    },
    "Action" : [ "mediastore:GetObject", "mediastore:DescribeObject" ],
    "Resource" : "arn:aws:mediastore:<region>:<owner acct number>:container/<container
 name>/*",
    "Condition" : {
      "Bool" : {
        "aws:SecureTransport" : [ "true", "false" ]
      }
    }
  } ]
}
```

範例容器政策：透過 HTTPS 的跨帳戶讀取存取

此範例政策允許存取`GetObject`和`DescribeObject`操作 (如資源路徑結尾的 * 所指定) 進行操作 (如資源路徑結尾的 * 所指定) 進行操作 <other acct number>。政策會指定此存取有其必要條件，也就是必須使用 HTTPS 進行操作：

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CrossAccountReadOverHttps",
      "Effect": "Allow",
      "Action": ["mediastore:GetObject", "mediastore:DescribeObject"],
      "Principal":{
        "AWS": "arn:aws:iam::<other acct number>:root"},
      "Resource": "arn:aws:mediastore:<region>:<owner acct number>:container/<container
 name>/*",
      "Condition": {
        "Bool": {
            "aws:SecureTransport": "true"
        }
      }
    }
  ]
}
```

範例容器政策：角色的跨帳戶讀取存取

此範例政策允許對任何由 <擁有者帳號> 所擁有的物件 (如資源路徑結尾的 * 所指定) 進行 GetObject 和 DescribeObject 操作存取。這允許 <其他帳號> 的任何使用者進行此存取 (如果該帳戶有擔任 <角色名稱> 所指定的角色)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CrossAccountRoleRead",
      "Effect": "Allow",
      "Action": ["mediastore:GetObject", "mediastore:DescribeObject"],
      "Principal":{
        "AWS": "arn:aws:iam::<other acct number>:role/<role name>"},
      "Resource": "arn:aws:mediastore:<region>:<owner acct number>:container/<container
 name>/*",
    }
  ]
}
```

範例容器政策：角色的跨帳戶完整存取

此範例政策允許跨帳戶存取來更新帳戶中的任何物件，只要使用者是透過 HTTP 登入即可。這也允許對已擔任指定角色的帳戶進行跨帳戶存取，透過 HTTP 或 HTTPS 來刪除、下載和描述物件：

- 第一個陳述式是 CrossAccountRolePostOverHttps。這允許對任何物件進行 PutObject 操作存取，並允許指定之帳戶的任何使用者進行此存取 (如果該帳戶有擔任 <角色名稱> 所指定的角色)。政策會指定此存取有其必要條件，也就是必須使用 HTTPS 進行操作 (提供 PutObject 存取權時，務必要包含此條件)。

  換言之，任何擁有跨帳戶存取權的委託人都可以存取 PutObject，但只能透過 HTTPS 進行。

- 第二個陳述式是 CrossAccountFullAccessExceptPost。這允許對任何物件進行除了 PutObject 以外的所有操作存取。這允許指定之帳戶的任何使用者進行此存取 (如果該帳戶有擔任 <角色名稱> 所指定的角色)。此存取沒有限定條件要求使用 HTTPS 進行操作。

  換言之，任何具有跨帳戶存取權的帳戶都可以進行 DeleteObject、GetObject 等存取 (但 PutObject 除外)，而且可以透過 HTTP 或 HTTPS 進行此存取。

  如果沒有將 PutObject 從第二個陳述式排除，則陳述式不會有效 (因為要是包含 PutObject，您就必須依條件明確設定 HTTPS)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CrossAccountRolePostOverHttps",
      "Effect": "Allow",
      "Action": "mediastore:PutObject",
      "Principal":{
        "AWS": "arn:aws:iam::<other acct number>:role/<role name>"},
      "Resource": "arn:aws:mediastore:<region>:<owner acct number>:container/<container
 name>/*",
      "Condition": {
        "Bool": {
            "aws:SecureTransport": "true"
        }
      }
    },
    {
      "Sid": "CrossAccountFullAccessExceptPost",
```

```
      "Effect": "Allow",
      "NotAction": "mediastore:PutObject",
      "Principal":{
        "AWS": "arn:aws:iam::<other acct number>:role/<role name>"},
      "Resource": "arn:aws:mediastore:<region>:<owner acct number>:container/<container
 name>/*"
    }
  ]
}
```

範例容器政策：限制存取特定 IP 地址

此範例政策允許存取所有 AWS ElementalMediaStore操作對指定容器中的物件進行操作。不過，要求必須源自於條件中所指定的 IP 地址範圍。

此陳述式中的條件會識別允許之 Internet Protocol Version 4 (IPv4) IP 地址的 198.51.100.* 範圍，但有一個例外：一九八

Condition 區塊使用 IpAddress 與 NotIpAddress 條件以及 aws:SourceIp 條件鍵，其為整個 AWS 的條件鍵。aws:sourceIp IPv4 值會使用標準 CIDR 表示法。如需詳細資訊，請參閱「」IP 地址條件運算子在 IAM 使用者指南中的。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AccessBySpecificIPAddress",
            "Effect": "Allow",
            "Action": [
                "mediastore:GetObject",
                "mediastore:DescribeObject"
            ],
            "Principal": "*",
            "Resource": "arn:aws:mediastore:<region>:<owner acct number>:container/
<container name>/*",
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": [
                        "198.51.100.0/24"
                    ]
                },
                "NotIpAddress": {
                    "aws:SourceIp": "198.51.100.188/32"
```

```
                }
            }
        }
    ]
}
```

AWS Elemental 中的跨來源資源分享 (CORS) 政策MediaStore

跨來源資源分享 (CORS) 會定義一種方式，讓載入單一個網域的用戶端 Web 應用程式，能與不同網域中的資源互動。藉助 AWS 元素中的 CORS 支持MediaStore，您可以使用MediaStore，並有選擇地允許跨源訪問MediaStore的費用。

> **Note**
>
> 如果您正在使用亞馬遜CloudFront要從具有 CORS 策略的容器中分發內容，請務必配置 AWS 元素的分配MediaStore（包括編輯緩存行為以設置 CORS 的步驟）。

本節提供 CORS 的概觀。副主題會說明如何使用 AWS Elemental 來啟用 CORSMediaStore控制台，或以編程方式使用MediaStoreREST API 和 AWS 軟件開發工具包。

主題

- CORS 使用案例情境
- 將 CORS 政策新增至容器
- 檢視 CORS 政策
- 編輯 CORS 政策
- 刪除 CORS 政策
- 對 CORS 問題進行故障診斷
- 範例 CORS 政策

CORS 使用案例情境

以下為使用 CORS 的情境範例：

- 方案 1：假設您正在分發 AWS Elemental 中的即時串流視訊MediaStore容器名稱為LiveVideo。您的使用者從特定來源 (如 `http://livevideo.mediastore.ap-southeast-2.amazonaws.com`) 載入資訊清單端點 `www.example.com`。您想要使用JavaScript視訊播放器，透過未經驗證的方式

來存取源自此容器的視訊。GET和PUT請求。瀏覽器通常會阻止JavaScript，但您可以在容器上設定 CORS 政策，明確地啟用這些來自www.example.com。

- 方案 2：假設您想要託管案例 1 中那個同樣來自MediaStore容器，但希望允許來自任何源的請求。您可設定 CORS 政策以允許使用萬用字元 (*) 來源，使任何來源的要求都能存取該視訊。

將 CORS 政策新增至容器

本節說明如何將跨來源資源分享 (CORS) 組態新增至 AWS ElementalMediaStore容器。CORS 允許載入同一個網域的用戶端 Web 應用程式，可以與不同網域中的資源互動。

若要設定容器以允許跨來源要求，請將 CORS 政策新增至容器。CORS 政策會定義一些規則，這些規則可識別您所允許可存取容器的來源、每個來源支援的操作 (HTTP 方法)，以及其他操作特定資訊。

當您將 CORS 政策新增至容器時，容器政策 (管理容器存取權利的政策) 照常適用。

若要新增 CORS 政策 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇您要為其建立 CORS 政策之容器的名稱。

   容器詳細資訊頁面隨即出現。
3. 在 Container CORS policy (容器 CORS 政策) 區段中，選擇 Create CORS policy (建立 CORS 政策)。
4. 依 JSON 格式插入政策，然後選擇 Save (儲存)。

若要新增 CORS 政策 (AWS CLI)

1. 建立檔案，此檔案會定義 CORS 政策：

```
[
  {
    "AllowedHeaders": [
      "*"
    ],
    "AllowedMethods": [
      "GET",
      "HEAD"
    ],
    "AllowedOrigins": [
```

```
      "*"
    ],
    "MaxAgeSeconds": 3000
  }
]
```

2. 在 AWS CLI 中，使用 `put-cors-policy` 命令。

```
aws mediastore put-cors-policy --container-name ExampleContainer --cors-policy
 file://corsPolicy.json --region us-west-2
```

此命令沒有傳回值。

檢視 CORS 政策

跨來源資源分享 (CORS) 會定義一種方式，讓載入單一個網域的用戶端 Web 應用程式，能與不同網域中的資源互動。

若要檢視 CORS 政策 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇您要檢視其 CORS 政策之容器的名稱。

   容器詳細資訊頁面隨即出現，並將 CORS 政策顯示在 Container CORS policy (容器 CORS 政策) 區段中。

若要檢視 CORS 政策 (AWS CLI)

- 在 AWS CLI 中，使用 `get-cors-policy` 命令：

```
aws mediastore get-cors-policy --container-name ExampleContainer --region us-west-2
```

以下範例顯示傳回值：

```
{
    "CorsPolicy": [
        {
            "AllowedMethods": [
                "GET",
```

```
                "HEAD"
            ],
            "MaxAgeSeconds": 3000,
            "AllowedOrigins": [
                "*"
            ],
            "AllowedHeaders": [
                "*"
            ]
        }
    ]
}
```

編輯 CORS 政策

跨來源資源分享 (CORS) 會定義一種方式，讓載入單一個網域的用戶端 Web 應用程式，能與不同網域中的資源互動。

若要編輯 CORS 政策 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇您要編輯其 CORS 政策之容器的名稱。

   容器詳細資訊頁面隨即出現。
3. 在 Container CORS policy (容器 CORS 政策) 區段中，選擇 Edit CORS policy (編輯 CORS 政策)。
4. 對政策進行變更，然後選擇 Save (儲存)。

若要編輯 CORS 政策 (AWS CLI)

1. 建立檔案，此檔案會定義更新的 CORS 政策：

```
[
  {
    "AllowedHeaders": [
      "*"
    ],
    "AllowedMethods": [
      "GET",
      "HEAD"
```

```
    ],
    "AllowedOrigins": [
      "https://www.example.com"
    ],
    "MaxAgeSeconds": 3000
  }
]
```

2. 在 AWS CLI 中，使用 `put-cors-policy` 命令。

```
aws mediastore put-cors-policy --container-name ExampleContainer --cors-policy
 file://corsPolicy2.json --region us-west-2
```

此命令沒有傳回值。

刪除 CORS 政策

跨來源資源分享 (CORS) 會定義一種方式，讓載入單一個網域的用戶端 Web 應用程式，能與不同網域中的資源互動。從容器刪除 CORS 政策會移除跨來源要求許可。

若要刪除 CORS 政策 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇您要刪除其 CORS 政策之容器的名稱。

   容器詳細資訊頁面隨即出現。
3. 在 Container CORS policy (容器 CORS 政策) 區段中，選擇 Delete CORS policy (刪除 CORS 政策)。
4. 選擇 Continue (繼續) 來確認，然後選擇 Save (儲存)。

若要刪除 CORS 政策 (AWS CLI)

- 在 AWS CLI 中，使用 `delete-cors-policy` 命令：

```
aws mediastore delete-cors-policy --container-name ExampleContainer --region us-
west-2
```

此命令沒有傳回值。

對 CORS 問題進行故障診斷

如果您在存取具有 CORS 政策的容器時遇到非預期行為，請依照下列步驟，對問題進行疑難排解。

1. 確認已將 CORS 政策附加至容器。

   如需指示，請參閱 the section called “檢視 CORS 政策”。

2. 使用您選擇的工具 (例如，瀏覽器的開發人員主控台)，擷取完整的要求和回應。確認附加至容器的 CORS 政策至少包含一個比對您的要求中資料的 CORS 規則，如下所示：

   a. 確認要求含有 `Origin` 標頭。

      如果缺少標頭，AWS ElementalMediaStore不會將請求視為跨來源請求，也不會在回應中傳送 CORS 回應標頭。

   b. 確認要求中的 `Origin` 標頭至少與特定 `AllowedOrigins` 的其中一個 `CORSRule` 元素比對相符。

      `Origin` 要求標頭中的配置、主機及連接埠值，都必須符合 `AllowedOrigins` 中的 `CORSRule`。例如，若將 `CORSRule` 設定為允許來源 `http://www.example.com`，則要求中的 `https://www.example.com` 及 `http://www.example.com:80` 來源，都不符合您組態中允許的來源。

   c. 確認要求中的 Method (若為預檢要求，則是 `Access-Control-Request-Method` 中指定的方法) 是相同 `AllowedMethods` 的其中一個 `CORSRule` 元素。

   d. 若是預檢要求，如果要求包含 `Access-Control-Request-Headers` 標頭，則請確認 `CORSRule` 對於 `AllowedHeaders` 標頭中的每個值，都包含了 `Access-Control-Request-Headers` 項目。

範例 CORS 政策

下列範例示範跨來源資源分享 (CORS) 政策。

主題

- 範例 CORS 政策：所有網域皆可的讀取存取
- 範例 CORS 政策：特定網域方可的讀取存取

範例 CORS 政策：所有網域皆可的讀取存取

下列政策允許任何網域的網頁從 AWS ElementalMediaStore容器。要求包含所有來自原始網域的 HTTP 標頭，而且服務只會回應來自原始網域的 HTTP GET 和 HTTP HEAD 要求。系統先將結果暫存於快取達 3,000 秒之後，再傳送一組新的結果。

```
[
  {
    "AllowedHeaders": [
      "*"
    ],
    "AllowedMethods": [
      "GET",
      "HEAD"
    ],
    "AllowedOrigins": [
      "*"
    ],
    "MaxAgeSeconds": 3000
  }
]
```

範例 CORS 政策：特定網域方可的讀取存取

下列政策允許的網頁從https://www.example.com從您的 AWS 元素中檢索內容MediaStore容器。要求包含所有來自 https://www.example.com 的 HTTP 標頭，而且服務只會回應來自 https://www.example.com 的 HTTP GET 和 HTTP HEAD 要求。系統先將結果暫存於快取達 3,000 秒之後，再傳送一組新的結果。

```
[
  {
    "AllowedHeaders": [
      "*"
    ],
    "AllowedMethods": [
      "GET",
      "HEAD"
    ],
    "AllowedOrigins": [
      "https://www.example.com"
    ],
    "MaxAgeSeconds": 3000
```

```
  }
]
```

AWS 元素中的物件生命週期政策MediaStore

對於每個容器，您可以建立物件生命週期政策，此政策可管理物件在容器中應存放的時間。當物件達到您指定的年齡上限後，AWS 元素MediaStore就會刪除物件。您可以刪除不再需要的物件以節省儲存成本。

您也可以指定 MediaStore 在物件到達一定存留期後，應將物件移動到不常存取儲存 (IA) 儲存體方案。存放在 IA 儲存體方案中的物件，其儲存和擷取的費率會和存放在標準儲存體方案中的物件不同。如需詳細資訊，請參閱 MediaStore 定價。

物件生命週期政策會包含規則，這些規則會依子資料夾規定物件的生命週期。(您無法將物件生命週期政策指派至個別物件)。您可以只將一個物件生命週期政策連接到容器，但您可以在每個物件生命週期政策中新增最多 10 個規則。如需更多詳細資訊，請參閱 物件生命週期政策的元件。

主題

物件生命週期政策的元件

物件生命週期政策會管理物件在 AWS 元素中保留的時間MediaStore容器。物件生命週期政策由一或多個規則組成，這些規則會規定物件的生命週期。規則可套用到一個資料夾、多個資料夾或者整個容器。

您可以只將一個物件生命週期政策連接到容器，且每個物件生命週期政策可包含最多 10 個規則。您無法將物件生命週期政策指派至個別物件。

物件生命週期政策中的規則

您可以建立三種規則類型：

- 刪除物件
- 生命週期轉換

暫時性資料

暫時性資料規則會設定物件在幾秒內過期。這種規則類型僅適用於在政策生效後新增到容器的物件。MediaStore 最多需要 20 分鐘的時間，才能將新的政策套用到容器。

暫時性資料的規則範例如下所示：

```
        {
            "definition": {
                "path": [ {"wildcard": "Football/index*.m3u8"} ],
                "seconds_since_create": [
                    {"numeric": [">", 120]}
                ]
            },
            "action": "EXPIRE"
        },
```

暫時性資料規則有三個部分：

- `path`：一律設為 `wildcard`。您可以使用此部分來定義要刪除的物件。您可以使用一或多個萬用字元，以星號 (*) 表示。每個萬用字元代表零個或多個字元的任何組合。例如，`"path": [ {"wildcard": "Football/index*.m3u8"} ]`，會套用到 `Football` 資料夾中符合 `index*.m3u8` 模式 (例如 index.m3u8、index1.m3us8 和 index123456.m3u8) 的所有檔案。單一規則最多可以包含 10 個 路徑。
- `seconds_since_create`：一律設為 `numeric`。您可指定介於 1-300 秒之間的值。您也可以將運算子設定為大於 (>) 或大於或等於 (>=)。
- `action`：一律設為 `EXPIRE`。

針對暫時性資料規則 (物件會在幾秒內過期)，在物件過期與刪除物件之間沒有延遲。

> **Note**
>
> `list-items` 回應中不會包含受暫時性資料規則限制的物件。此外，由於臨時數據規則而過期的對象不會發出CloudWatch事件，當它們到期時。

刪除物件

刪除物件規則會將物件設定為在幾天內過期。這種規則類型適用於容器中的所有物件，即使它們在建立原則之前已新增至容器。MediaStore 最多需要 20 分鐘的時間才能套用新的政策，但物件最多可能需要 24 小時的時間才能從容器中清除。

刪除物件的兩個規則範例如下所示：

```
        {
            "definition": {
                "path": [ { "prefix": "FolderName/" }  ],
                "days_since_create": [
                    {"numeric": [">" , 5]}
                ]
            },
            "action": "EXPIRE"
        },
        {
            "definition": {
                "path": [ { "wildcard": "Football/*.ts" }  ],
                "days_since_create": [
                    {"numeric": [">" , 5]}
                ]
            },
            "action": "EXPIRE"
        }
```

刪除物件規則有三個部分：

- `path`：設為 `prefix` 或 `wildcard`。您不可在相同規則中混合使用 `prefix` 和 `wildcard`。如果要同時使用兩者，則必須為 `prefix` 建立一個規則，並為 `wildcard` 建立另一個規則，如上述範例所示。
  - `prefix` - 如果您想要刪除特定資料夾內的所有物件，可以將路徑設為 `prefix`。如果參數是空的 (`"path": [ { "prefix": "" } ],`)，目標就是存放在目前容器內任何位置的所有物件。單一規則最多可以包含 10 個 `prefix` 路徑。
  - `wildcard` - 如果您要根據檔案名稱及/或檔案類型刪除特定物件，則可將路徑設為 `wildcard`。您可以使用一或多個萬用字元，以星號 (*) 表示。每個萬用字元代表零個或多個字元的任何組合。例如，`"path": [ {"wildcard": "Football/*.ts"} ],` 適用於 `Football` 資料夾中符合 `*.ts` 模式的所有檔案 (例如 filename.ts、filename1.ts，以及 filename123456.ts)。單一規則最多可以包含 10 個 `wildcard` 路徑。

- days_since_create：一律設為 numeric。您可指定介於 1-36,500 天之間的值。您也可以將運算子設定為大於 (>) 或大於或等於 (>=)。
- action：一律設為 EXPIRE。

若為刪除物件規則 (物件會在幾天內過期)，在物件過期與刪除物件之間可能會有些許延遲。不過，只要物件過期，帳單也會立即變更。例如，如果生命週期規則指定 10 days_since_create，在物件達到第 10 天後，此帳戶就不會針對此物件收費，即使尚未刪除此物件。

生命週期轉換

生命週期轉換規則會將物件設為在到達一定存留期後移動到不常存取 (IA) 儲存體方案 (以天數計)。存放在 IA 儲存體方案中的物件，其儲存和擷取的費率會和存放在標準儲存體方案中的物件不同。如需詳細資訊，請參閱 MediaStore 定價。

一旦物件移動到 IA 儲存體方案，您就無法將其移回標準儲存體方案。

生命週期轉換規則適用於容器中的所有物件，即使它們在建立原則之前已新增至容器。MediaStore 最多需要 20 分鐘的時間才能套用新的政策，但物件最多可能需要 24 小時的時間才能從容器中清除。

生命週期轉移規則的範例如下所示：

```
{
    "definition": {
        "path": [
            {"prefix": "AwardsShow/"}
        ],
        "days_since_create": [
            {"numeric": [">=" , 30]}
        ]
    },
    "action": "ARCHIVE"
}
```

生命週期轉換規則有三個部分：

- path：設為 prefix 或 wildcard。您不可在相同規則中混合使用 prefix 和 wildcard。如果要同時使用兩者，您必須為 prefix 建立一個規則，並為 wildcard 建立單獨建立另一個規則。
  - prefix – 如果您希望將特定資料夾內的所有物件移動到 IA 儲存體方案，您可以將路徑設為 prefix。如果參數為空白 ("path": [{ "prefix": "" }],)，則目標就會是儲存在目前容器中任何位置的所有物件。單一規則最多可以包含 10 個 prefix 路徑。

- wildcard – 如果您希望根據檔案名稱和 (或) 檔案類型，將特定物件移動到 IA 儲存體方案，您可以將路徑設為 wildcard。您可以使用一或多個萬用字元，以星號 (*) 表示。每個萬用字元代表零個或多個字元的任何組合。例如，"path": [{"wildcard": "Football/*.ts"}], 適用於 Football 資料夾中符合 *.ts 模式的所有檔案 (例如 filename.ts、filename1.ts，以及 filename123456.ts)。單一規則最多可以包含 10 個 wildcard 路徑。
- days_since_create：一律設為 "numeric": [">=" , 30]。
- action：一律設為 ARCHIVE。

範例

例如，名為 LiveEvents 的容器有四個子資料夾：Football、Baseball、Basketball 和 AwardsShow。指派給 LiveEvents 資料夾的物件生命週期政策可能如下所示：

```
{
    "rules": [
         {
            "definition": {
                "path": [
                    {"prefix": "Football/"},
                    {"prefix": "Baseball/"}
                ],
                "days_since_create": [
                    {"numeric": [">" , 28]}
                ]
            },
            "action": "EXPIRE"
        },
        {
            "definition": {
                "path": [ { "prefix": "AwardsShow/" }  ],
                "days_since_create": [
                    {"numeric": [">=" , 15]}
                ]
            },
            "action": "EXPIRE"
        },
        {
            "definition": {
                "path": [ { "prefix": "" }  ],
                "days_since_create": [
                    {"numeric": [">" , 40]}
```

```
                ]
            },
            "action": "EXPIRE"
        },
        {
            "definition": {
                "path": [ { "wildcard": "Football/*.ts" }  ],
                "days_since_create": [
                    {"numeric": [">" , 20]}
                ]
            },
            "action": "EXPIRE"
        },
        {
            "definition": {
                "path": [
                    {"wildcard": "Football/index*.m3u8"}
                ],
                "seconds_since_create": [
                    {"numeric": [">" , 15]}
                ]
            },
            "action": "EXPIRE"
        },
        {
                "definition": {
                    "path": [
                        {"prefix": "Program/"}
                    ],
                    "days_since_create": [
                        {"numeric": [">=" , 30]}
                    ]
                },
                "action": "ARCHIVE"
            }
    ]
}
```

上述政策會指定以下項目：

- 第一條規則指示 AWS 元素MediaStore來刪除存放在`LiveEvents/Football`資料夾和`LiveEvents/Baseball`文件夾之後，它們超過 28 天。

- 第二個規則會指示此服務在 LiveEvents/AwardsShow 資料夾中存放的物件年齡為 15 天以上後，將其刪除。
- 第三個規則會指示此服務在 LiveEvents 容器中任何位置存放的物件年齡超過 40 天後，將其刪除。這個規則適用於直接在 LiveEvents 容器中存放的物件，以及在容器的四個子資料夾中任一項存放的物件。
- 第四個規則會服務在 Football 資料夾中符合模式 *.ts 的物件超過 20 天後，將其刪除。
- 第五條規則指示服務刪除Football與模式相符的資料夾index*.m3u8之後，他們超過 15 秒。MediaStore將這些文件放在容器中 16 秒後刪除這些文件。
- 第六個規則會指示服務在物件存留期超過 30 天後，將 Program 資料夾中的物件移動到 IA 儲存體方案。

如需物件生命週期政策的更多範例，請參閱 範例物件生命週期政策。

將物件生命週期政策新增至容器

物件生命週期政策可讓您指定要將物件存放在容器多久。您會設定過期日期，且在過期日期之後，AWS 元素MediaStore就會刪除物件。此服務最多需要 20 分鐘的時間，才能將新政策套用至容器。

如需有關如何建構生命週期政策的資訊，請參閱 物件生命週期政策的元件。

> **Note**
>
> 若為刪除物件規則 (物件會在幾天內過期)，在物件過期與刪除物件之間可能會有些許延遲。不過，只要物件過期，帳單也會立即變更。例如，如果生命週期規則指定 10 days_since_create，在物件達到第 10 天後，此帳戶就不會針對此物件收費，即使尚未刪除此物件。

新增物件生命週期政策 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇您要為其建立物件生命週期政策之容器的名稱。

   容器詳細資訊頁面隨即出現。
3. 在 Object lifecycle policy (物件生命週期政策) 區段中，選擇 Create object lifecycle policy (建立物件生命週期政策)。

4. 依 JSON 格式插入政策，然後選擇 Save (儲存)。

新增物件生命週期政策 (AWS CLI)

1. 建立檔案，此檔案會定義物件生命週期政策：

```
{
    "rules": [
         {
            "definition": {
                "path": [
                    {"prefix": "Football/"},
                    {"prefix": "Baseball/"}
                ],
                "days_since_create": [
                    {"numeric": [">" , 28]}
                ]
            },
            "action": "EXPIRE"
        },
        {
            "definition": {
                "path": [
                    {"wildcard": "AwardsShow/index*.m3u8"}
                ],
                "seconds_since_create": [
                    {"numeric": [">" , 8]}
                ]
            },
            "action": "EXPIRE"
        }
    ]
}
```

2. 在 AWS CLI 中，使用 `put-lifecycle-policy` 命令：

```
aws mediastore put-lifecycle-policy --container-name LiveEvents --lifecycle-
policy file://LiveEventsLifecyclePolicy.json --region us-west-2
```

此命令沒有傳回值。此服務會將指定政策連接至容器。

檢視物件生命週期政策

物件生命週期政策會指定物件在容器中應保留的時間。

檢視物件生命週期政策 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇您要檢視其物件生命週期政策之容器的名稱。

   容器詳細資訊頁面隨即出現，物件生命週期政策顯示在 Object lifecycle policy (物件生命週期政策) 區段中。

查看物件生命週期政策 (AWS CLI)

- 在 AWS CLI 中，使用 `get-lifecycle-policy` 命令：

```
aws mediastore get-lifecycle-policy --container-name LiveEvents --region us-west-2
```

以下範例顯示傳回值：

```
{
    "LifecyclePolicy": "{
        "rules": [
            {
                "definition": {
                    "path": [
                        {"prefix": "Football/"},
                        {"prefix": "Baseball/"}
                    ],
                    "days_since_create": [
                        {"numeric": [">" , 28]}
                    ]
                },
             "action": "EXPIRE"
            }
        ]
    }"
}
```

編輯物件生命週期政策

您無法編輯現有的物件生命週期政策。不過，您可以透過上傳取代政策來變更現有的政策。此服務最多需要 20 分鐘的時間，才能將更新的政策套用至容器。

編輯物件生命週期政策 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇您要編輯其物件生命週期政策之容器的名稱。

   容器詳細資訊頁面隨即出現。
3. 在 Object lifecycle policy (物件生命週期政策) 區段中，選擇 Edit object lifecycle policy (編輯物件生命週期政策)。
4. 對政策進行變更，然後選擇 Save (儲存)。

編輯物件生命週期政策 (AWS CLI)

1. 建立檔案，此檔案會定義更新的物件生命週期政策：

```
{
    "rules": [
         {
            "definition": {
                "path": [
                    {"prefix": "Football/"},
                    {"prefix": "Baseball/"}
                    {"prefix": "Basketball/"}
                ],
                "days_since_create": [
                    {"numeric": [">" , 28]}
                ]
            },
            "action": "EXPIRE"
        }
    ]
}
```

2. 在 AWS CLI 中，使用 `put-lifecycle-policy` 命令：

```
aws mediastore put-lifecycle-policy --container-name LiveEvents --lifecycle-
policy file://LiveEvents2LifecyclePolicy --region us-west-2
```

此命令沒有傳回值。此服務會將指定政策連接至容器，取代先前的政策。

刪除物件生命週期政策

刪除物件生命週期政策時，此服務最多需要 20 分鐘的時間，才能將變更套用至容器。

刪除物件生命週期政策 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇您要刪除其物件生命週期政策之容器的名稱。

   容器詳細資訊頁面隨即出現。
3. 在 Object lifecycle policy (物件生命週期政策) 區段中，選擇 Delete lifecycle policy (刪除生命週期政策)。
4. 選擇 Continue (繼續) 來確認，然後選擇 Save (儲存)。

刪除物件生命週期政策 (AWS CLI)

- 在 AWS CLI 中，使用 `delete-lifecycle-policy` 命令：

```
aws mediastore delete-lifecycle-policy --container-name LiveEvents --region us-
west-2
```

此命令沒有傳回值。

範例物件生命週期政策

下列範例顯示物件生命週期政策。

主題

- 範例物件生命週期政策：在幾秒鐘內過期
- 範例物件生命週期政策：在幾天內過期
- 範例物件生命週期政策：轉換至不常存取的存取體方案

- 範例物件生命週期政策：多重規則
- 範例物件生命週期政策：清空容器

範例物件生命週期政策：在幾秒鐘內過期

以下政策會指定 MediaStore 刪除符合下列所有條件的物件：

- 物件是在政策生效後新增到容器的。
- 物件存放在 Football 資料夾中。
- 物件具有副檔名 m3u8。
- 物件已在容器中超過 20 秒。

```
{
    "rules": [
        {
            "definition": {
                "path": [
                    {"wildcard": "Football/*.m3u8"}
                ],
                "seconds_since_create": [
                    {"numeric": [ ">", 20 ]}
                ]
            },
            "action": "EXPIRE"
        }
    ]
}
```

範例物件生命週期政策：在幾天內過期

以下政策會指定 MediaStore 刪除符合下列所有條件的物件：

- 物件存放在 Program 資料夾中
- 物件具有副檔名 ts
- 物件已在容器中超過 5 天

```
{
```

```
    "rules": [
        {
            "definition": {
                "path": [
                    {"wildcard": "Program/*.ts"}
                ],
                "days_since_create": [
                    {"numeric": [ ">", 5 ]}
                ]
            },
            "action": "EXPIRE"
        }
    ]
}
```

範例物件生命週期政策：轉換至不常存取的存取體方案

以下政策會指定 MediaStore 在物件存留期超過 30 天後，將物件移動到不常存取 (IA) 儲存體方案。存放在 IA 儲存體方案中的物件，其儲存和擷取的費率會和存放在標準儲存體方案中的物件不同。

`days_since_create` 欄位必須設為 `"numeric": [">=" ,30]`。

```
{
    "rules": [
        {
            "definition": {
                "path": [
                    {"prefix": "Football/"},
                    {"prefix": "Baseball/"}
                ],
                "days_since_create": [
                    {"numeric": [">=" , 30]}
                ]
            },
            "action": "ARCHIVE"
        }
    ]
}
```

範例物件生命週期政策：多重規則

以下政策會指定 MediaStore 執行下列作業：

- 在物件存留期達 30 天後，將存放在 AwardsShow 資料夾中的物件移動到不常存取 (IA) 儲存體方案。
- 刪除副檔名為 m3u8 並且存放在 Football 資料夾中超過 20 秒的物件。
- 刪除存放在 April 資料夾中超過 10 天的物件。
- 刪除副檔名為 ts，且存放在 Program 資料夾中超過 5 天的物件。

```
{
    "rules": [
        {
            "definition": {
                "path": [
                    {"prefix": "AwardsShow/"}
                ],
                "days_since_create": [
                    {"numeric": [ ">=" , 30 ]}
                ]
            },
            "action": "ARCHIVE"
        },
        {
            "definition": {
                "path": [
                    {"wildcard": "Football/*.m3u8"}
                ],
                "seconds_since_create": [
                    {"numeric": [ ">", 20 ]}
                ]
            },
            "action": "EXPIRE"
        },
        {
            "definition": {
                "path": [
                    {"prefix": "April"}
                ],
                "days_since_create": [
                    {"numeric": [ ">", 10 ]}
                ]
            },
            "action": "EXPIRE"
        },
```

```
        {
            "definition": {
                "path": [
                    {"wildcard": "Program/*.ts"}
                ],
                "days_since_create": [
                    {"numeric": [ ">", 5 ]}
                ]
            },
            "action": "EXPIRE"
        }
    ]
}
```

範例物件生命週期政策：清空容器

以下物件生命週期政策會指定 MediaStore 在物件新增到容器的 1 天後刪除容器內的所有物件 (包括資料夾和子資料夾)。如果容器在套用這項政策之前保留了任何物件，MediaStore會在政策生效 1 天後刪除物件。此服務最多需要 20 分鐘的時間，才能將新政策套用至容器。

```
{
    "rules": [
        {
            "definition": {
                "path": [
                    {"wildcard": "*"}
                ],
                "days_since_create": [
                    {"numeric": [ ">=", 1 ]}
                ]
            },
            "action": "EXPIRE"
        }
    ]
}
```

AWS Elemental 中的指標政策 MediaStore

對於每個容器，您可以新增指標政策，以允許 AWS Elemental 將指標傳送 MediaStore 到亞馬遜 CloudWatch。新的政策需要最多 20 分鐘就會生效。如需每個 MediaStore 量度的說明，請參閱MediaStore 指標。

指標政策包含下列項目：

- 在容器層級啟用或停用指標的設定。
- 從零到五之間任意數量的規則，用在物件層級啟用指標。如果政策包含規則，則每個規則都必須包含下列兩項：
  - 一個物件群組，定義要包括在群組中的物件。定義可以是路徑或檔案名稱，但不能超過 900 個字元。有效字元包括：a-z、A-Z、0-9、_ (底線)、= (等號)、: (冒號)、. (句號)、- (連字號)、~ (波浪號)、/ (正斜線) 和 * (星號)。可接受萬用字元 (*)。
  - 允許您參照物件群組的物件群組名稱。名稱不能超過 30 個字元。有效字元為 a-z、A-Z、0-9 和 _ (底線)。

如果物件符合多個規則， CloudWatch 會顯示每個相符規則的資料點。例如，如果一個物件符合兩個名為`rule1` AND 的規則`rule2`，則 CloudWatch 會顯示這些規則的兩個資料點。第一個具有 `ObjectGroupName=rule1` 的維度，第二個具有 `ObjectGroupName=rule2` 的維度。

主題

新增指標政策

指標政策包含指定 AWS Elemental MediaStore 傳送給亞馬遜的指標的規則 CloudWatch。如需指標政策的範例，請參閱指標政策範例。

新增指標政策 (主控台)

1. 開啟主 MediaStore 控台，網址為 https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇您要為其新增指標政策之容器的名稱。

   容器詳細資訊頁面隨即出現。
3. 在 Metric policy (指標政策) 區段中，選擇 Create metric policy (建立指標政策)。
4. 依 JSON 格式插入政策，然後選擇 Save (儲存)。

檢視指標政策

您可以使用主控台或 AWS CLI 來檢視容器的指標政策。

檢視指標政策 (主控台)

1. 開啟主 MediaStore 控台，網址為 https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇容器名稱。

   容器詳細資訊頁面隨即出現。政策會顯示在 Metric policy (指標政策) 區段中。

編輯指標政策

指標政策包含指定 AWS Elemental MediaStore 傳送給亞馬遜的指標的規則 CloudWatch。當您編輯現有的指標政策時，新政策需要長達 20 分鐘才會生效。如需指標政策的範例，請參閱指標政策範例。

編輯指標政策 (主控台)

1. 開啟主 MediaStore 控台，網址為 https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇容器名稱。
3. 在 Metric policy (指標政策) 區段中，選擇 Edit metric policy (編輯指標政策)。
4. 進行適當變更，然後選擇 Save (儲存)。

指標政策範例

下列範例示範針對不同使用案例所建構的指標政策。

主題

- 指標政策範例：容器層級指標
- 指標政策範例：路徑層級指標
- 指標政策範例：容器層級和路徑層級指標
- 指標政策範例：使用萬用字元的路徑層級指標
- 指標政策範例：具有重疊規則的路徑層級指標

指標政策範例：容器層級指標

此範例政策指出 AWS Elemental MediaStore 應該在容器層級將指標傳送 CloudWatch 至亞馬遜。例如，這包括 RequestCount 指標，此指標會計算對容器提出的 Put 請求數目。或者，您可以將其設定為 DISABLED。

由於此原則中沒有規則，因此 MediaStore 不會在路徑層級傳送量度。例如，您看不到對此容器內的特定資料夾提出的 Put 請求數。

```
{
   "ContainerLevelMetrics": "ENABLED"
}
```

指標政策範例：路徑層級指標

此範例政策指出 AWS Elemental 不 MediaStore 應在容器層級將指標傳送 CloudWatch 至亞馬遜。此外， MediaStore 應該為兩個特定資料夾中的物件傳送指標： baseball/saturday 和 football/saturday。 MediaStore 請求的指標如下：

- 對baseball/saturday資料夾的要求的 CloudWatch 維度為ObjectGroupName=baseballGroup。
- football/saturday 資料夾的請求具有維度 ObjectGroupName=footballGroup。

```
{
   "ContainerLevelMetrics": "DISABLED",
   "MetricPolicyRules": [
     {
       "ObjectGroup": "baseball/saturday",
       "ObjectGroupName": "baseballGroup"
     },
     {
       "ObjectGroup": "football/saturday",
       "ObjectGroupName": "footballGroup"
     }
   ]
}
```

指標政策範例：容器層級和路徑層級指標

此範例政策指出 AWS Elemental MediaStore 應該在容器層級將指標傳送 CloudWatch 至亞馬遜。此外， MediaStore 應該傳送兩個特定資料夾中物件的量度：`baseball/saturday`和`football/saturday`。 MediaStore 請求的指標如下：

- 對`baseball/saturday`資料夾的要求的 CloudWatch 維度為`ObjectGroupName=baseballGroup`。
- 對`football/saturday`資料夾的要求具有 CloudWatch 維度`ObjectGroupName=footballGroup`。

```
{
   "ContainerLevelMetrics": "ENABLED",
   "MetricPolicyRules": [
     {
       "ObjectGroup": "baseball/saturday",
       "ObjectGroupName": "baseballGroup"
     },
     {
       "ObjectGroup": "football/saturday",
       "ObjectGroupName": "footballGroup"
     }
   ]
}
```

指標政策範例：使用萬用字元的路徑層級指標

此範例政策指出 AWS Elemental MediaStore 應該在容器層級將指標傳送 CloudWatch 至亞馬遜。此外，還 MediaStore 應該根據物件的檔案名稱傳送物件的度量。萬用字元表示物件可以儲存在容器中的任何位置，而且它們可以有任何檔案名稱，只要它以 `.m3u8` 副檔名結尾即可。

```
{
   "ContainerLevelMetrics": "ENABLED",
   "MetricPolicyRules": [
     {
       "ObjectGroup": "*.m3u8",
       "ObjectGroupName": "index"
     }
   ]
 }
```

指標政策範例：具有重疊規則的路徑層級指標

此範例政策指出 AWS Elemental MediaStore 應該在容器層級將指標傳送 CloudWatch 至亞馬遜。此外， MediaStore 應該傳送兩個資料夾的量度：`sports/football/saturday`和`sports/football`。

`sports/football/saturday`資料夾 MediaStore 要求的量度具有的 CloudWatch 維度`ObjectGroupName=footballGroup1`。由於儲存在 `sports/football` 資料夾中的物件符合這兩個規則，因此 CloudWatch 會顯示這些物件的兩個資料點：一個具有 `ObjectGroupName=footballGroup2` 的維度，第二個具有 `ObjectGroupName=footballGroup1` 的維度。

```
{
   "ContainerLevelMetrics": "ENABLED",
   "MetricPolicyRules": [
     {
       "ObjectGroup": "sports/football/saturday",
       "ObjectGroupName": "footballGroup1"
     },
     {
       "ObjectGroup": "sports/football",
       "ObjectGroupName": "footballGroup2"
     }
   ]
 }
```

AWS Elemental 中的資料夾MediaStore

資料夾是容器內的劃分區。您使用資料夾細分容器，其方式與您建立子資料夾來劃分檔案系統中的資料夾相同。您可以建立最多 10 個層級的資料夾 (不包括容器本身)。

資料夾可選用；您可以選擇直接上傳物件給容器，而不是資料夾。不過，資料夾不失為組織物件的簡便方法。

若要將物件上傳至資料夾，請指定資料夾的路徑。如果資料夾已經存在，AWS ElementalMediaStore會將物件存放在資料夾中。如果資料夾不存在，則服務會建立此資料夾，再將物件存放在其中。

例如，假設您有一個名為`movies`，然後上傳一個名為`mlaw.ts`使用路徑`premium/canada`。AWS ElementalMediaStore將物件存放在 premium 資料夾底下的 canada 子資料夾中。如果這兩個資料夾都不存在，服務就會同時建立 `premium` 資料夾和 `canada` 子資料夾，然後將您的物件存放在 `canada` 子資料夾中。如果您只有指定 `movies` 容器 (沒有路徑)，服務則直接將物件存放在容器中。

AWS ElementalMediaStore當您刪除資料夾中的最後一個物件時， 會自動刪除該資料夾。服務也會刪除任何在該資料夾上層的空資料夾。例如，假設您有一個不含任何檔案、名為 premium 的資料夾，但其中包含一個名為 `canada` 的子資料夾。`canada` 子資料夾包含一個名為 `mlaw.ts` 的檔案。如果您刪除檔案 `mlaw.ts`，則服務會同時刪除 `premium` 和 `canada` 資料夾。此自動刪除動作僅適用於資料夾。服務並不會刪除空的容器。

主題

資料夾名稱規則

為資料夾選擇名稱時，請記得下列項目：

- 名稱只能包含以下字元：大寫字母 (A-Z)、小寫字母 (a-z)、數字 (0-9)、句點 (.)、連字號 (.)、連字號 (-)、波浪號 (~)、底線 (_)、等於符號 (=) 和冒號 (:)。
- 此名稱至少必須有一個字元。空文件夾名稱（ 如`folder1//folder3/` ）不允許。
- 名稱區分大小寫。例如，您在同一個容器或資料夾中可以有名為 `myFolder` 的資料夾以及名為 `myfolder` 的資料夾，因為這些名稱是唯一的。

- 此名稱在其父系容器或資料夾內必須是唯一的。例如，您可以在兩個不同的容器 (myfolder 和 movies/myfolder) 中各建立一個名為 sports/myfolder 的資料夾。
- 此名稱可以與其父容器相同。
- 不可在建立資料夾後加以重新命名。

建立資料夾

您可以在上傳物件時建立資料夾。若要將物件上傳至資料夾，請指定資料夾的路徑。如果資料夾已經存在，AWS ElementalMediaStore會將物件存放在資料夾中。如果資料夾不存在，則服務會建立此資料夾，再將物件存放在其中。

如需詳細資訊，請參閱 the section called “上傳物件”。

刪除資料夾

只有在資料夾是空的時，您才可以刪除資料夾；您無法刪除含有物件的資料夾。

AWS ElementalMediaStore當您刪除資料夾中的最後一個物件時， 會自動刪除該資料夾。服務也會刪除任何在該資料夾上層的空資料夾。例如，假設您有一個不含任何檔案、名為 premium 的資料夾，但其中包含一個名為 canada 的子資料夾。canada 子資料夾包含一個名為 mlaw.ts 的檔案。如果您刪除檔案 mlaw.ts，則服務會同時刪除 premium 和 canada 資料夾。此自動刪除動作僅適用於資料夾。服務並不會刪除空的容器。

如需詳細資訊，請參閱刪除物件。

AWS Elemental 中的物件MediaStore

AWS ElementalMediaStore資產即稱為對象。您可以將物件上傳至容器或容器中的資料夾。

在 MediaStore 中，您可以上傳、下載和刪除物件：

- 上傳 – 將物件新增至容器或資料夾。這與建立物件有所不同。您必須先在本機建立物件，然後才能將這些物件上傳至 MediaStore。
- 下載 – 將物件從 MediaStore 複製到其他位置。這不會從 MediaStore 移除物件。
- 刪除 – 從 MediaStore 完全移除物件。您可以個別刪除物件，或者可以新增物件生命週期政策，來自動在指定的持續時間後刪除容器中的物件。

MediaStore 接受所有檔案類型。

主題

上傳物件

您可以將物件上傳至容器，或容器中的資料夾。若要將物件上傳至資料夾，請指定資料夾的路徑。如果資料夾已經存在，AWS ElementalMediaStore將物件存放在資料夾中。如果資料夾不存在，則服務會建立此資料夾，再將物件存放在其中。如需資料夾的詳細資訊，請參閱AWS Elemental 中的資料夾MediaStore。

您可以使用 MediaStore 主控台或 AWS CLI 來上傳物件。

MediaStore 支援物件的區塊傳輸，這可透過將物件設為在上傳期間中仍可供下載來減少延遲。若要使用此功能，請將物件的上傳可用性設為 `streaming`。您可以在使用 API 上傳物件時，設定此標頭的值。如果您沒有在請求中指定此標頭，MediaStore 會指派預設值 `standard` 做為物件的上傳可用性。

標準上傳可用性的物件大小不可超過 25 MB，串流上傳可用性的物件大小限制為 10 MB。

> **Note**
>
> 物件資料名稱只能包含字母、數字、句點 (.)、底線 (_)、波狀符號 (~)、連字號 (-)、等號 (=) 和冒號 (:)。

若要上傳物件 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇容器的名稱。容器的詳細資訊面板隨即出現。
3. 選擇 Upload object (上傳物件)。
4. 在 Target path (目標路徑) 中，輸入資料夾的路徑。例如： `premium/canada`。如果您所指定的路徑中有任何資料夾尚不存在，服務會自動建立該資料夾。
5. 在 Object (物件) 區段中，選擇 Browse (瀏覽)。
6. 導覽至適當的資料夾，然後選擇一個物件來上傳。
7. 選擇 Open (開啟)，然後選擇 Upload (上傳)。

> **Note**
>
> 如果名稱相同的檔案已經存在於選取的資料夾，服務會將原始檔案取代為上傳的檔案。

若要上傳物件 (AWS CLI)

- 在 AWS CLI 中，使用 `put-object` 命令。您也可以包含下列任何參數：`content-type`、`cache-control` (允許呼叫者控制物件的快取行為) 以及 `path` (將物件放在容器的某個資料夾中)。

> **Note**
>
> 上傳物件後，就無法編輯 `content-type`、`cache-control` 或 `path`。

```
aws mediastore-data put-object --endpoint https://
aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com --body README.md --path /
```

```
folder_name/README.md --cache-control "max-age=6, public" --content-type binary/
octet-stream --region us-west-2
```

以下範例顯示傳回值：

```
{
    "ContentSHA256":
 "74b5fdb517f423ed750ef214c44adfe2be36e37d861eafe9c842cbe1bf387a9d",
    "StorageClass": "TEMPORAL",
    "ETag": "af3e4731af032167a106015d1f2fe934e68b32ed1aa297a9e325f5c64979277b"
}
```

檢視物件的清單

您可以使用 AWS 元素MediaStore主控台來檢視存放在容器最上層或在資料夾中的項目 (物件和資料夾)。不會顯示存放在目前容器或資料夾之子資料夾中的項目。無論容器中有多少資料夾或子資料夾，您都可以使用 AWS CLI 來檢視容器中的物件及資料夾清單。

若要檢視特定容器中的物件清單 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇有您所要檢視之資料夾的容器的名稱。
3. 從清單中選擇資料夾的名稱。

   詳細資訊頁面隨即出現，並顯示所有存放在該資料夾中的資料夾及物件。

若要檢視特定資料夾中的物件清單 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇有您所要檢視之資料夾的容器的名稱。

   詳細資訊頁面隨即出現，並顯示所有存放在該容器中的資料夾及物件。

若要檢視特定容器中的物件及資料夾清單 (AWS CLI)

- 在 AWS CLI 中，使用 `list-items` 命令：

```
aws mediastore-data list-items --endpoint https://
aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com --region us-west-2
```

以下範例顯示傳回值：

```
{
    "Items": [
        {
            "ContentType": "image/jpeg",
            "LastModified": 1563571859.379,
            "Name": "filename.jpg",
            "Type": "OBJECT",
            "ETag":
 "543ab21abcd1a234ab123456a1a2b12345ab12abc12a1234abc1a2bc12345a12",
            "ContentLength": 3784
        },
        {
            "Type": "FOLDER",
            "Name": "ExampleLiveDemo"
        }
    ]
}
```

> **Note**
>
> 受 seconds_since_create 規則限制的物件不會包含在 list-items 回應中。

若要檢視特定資料夾中的物件及資料夾清單 (AWS CLI)

- 在 AWS CLI 中，使用 list-items 命令，並在請求的結尾包含指定的資料夾名稱：

```
aws mediastore-data list-items --endpoint https://
aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com --path /folder_name --
region us-west-2
```

以下範例顯示傳回值：

```
{
```

```
    "Items": [
        {
            "Type": "FOLDER",
            "Name": "folder_1"
        },
        {
            "LastModified": 1563571940.861,
            "ContentLength": 2307346,
            "Name": "file1234.jpg",
            "ETag":
 "111a1a22222a1a1a222abc333a444444b55ab1111ab2222222222ab333333a2b",
            "ContentType": "image/jpeg",
            "Type": "OBJECT"
        }
    ]
}
```

> **Note**
>
> 受 `seconds_since_create` 規則限制的物件不會包含在 `list-items` 回應中。

檢視物件的詳細資訊

上傳數據元後，AWS 元素MediaStore會存放一些詳細資訊，例如修改日期、內容長度、ETag (實體標籤) 和內容類型。若要了解如何使用物件的中繼資料，請參閱MediaStore與 HTTP 快取的互動。

若要檢視物件的詳細資訊 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇有您所要檢視之物件的容器的名稱。
3. 如果您想要檢視的物件是在某個資料夾中，請繼續選擇資料夾名稱，直到您看到該物件為止。
4. 選擇物件的名稱。

   詳細資訊頁面隨即出現，並顯示物件的相關資訊。

若要檢視物件的詳細資訊 (AWS CLI)

- 在 AWS CLI 中，使用 `describe-object` 命令：

```
aws mediastore-data describe-object --endpoint https://
aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com --path /folder_name/
file1234.jpg --region us-west-2
```

以下範例顯示傳回值：

```
{
    "ContentType": "image/jpeg",
    "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
    "ContentLength": "2307346",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
}
```

下載物件

您可以使用主控台來下載物件。您可以使用 AWS CLI 來下載一個物件，或是僅下載物件的一部分。

若要下載物件 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇有您所要下載之物件的容器的名稱。
3. 如果您想要下載的物件是在某個資料夾中，請繼續選擇資料夾名稱，直到您看到該物件為止。
4. 選擇物件的名稱。
5. 在 Object details (物件詳細資訊) 頁面上，選擇 Download (下載)。

若要下載物件 (AWS CLI)

- 在 AWS CLI 中，使用 `get-object` 命令：

```
aws mediastore-data get-object --endpoint https://
aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com --path=/folder_name/
README.md README.md --region us-west-2
```

以下範例顯示傳回值：

```
{
    "ContentLength": "2307346",
```

```
    "ContentType": "image/jpeg",
    "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
    "StatusCode": 200
}
```

若要下載物件的一部分 (AWS CLI)

- 在 AWS CLI 中，使用 `get-object` 命令，並指定範圍。

```
aws mediastore-data get-object --endpoint https://
aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com --path /folder_name/
README.md --range="bytes=0-100" README2.md --region us-west-2
```

以下範例顯示傳回值：

```
{
    "StatusCode": 206,
    "ContentRange": "bytes 0-100/2307346",
    "ContentLength": "101",
    "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
    "ContentType": "image/jpeg",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
}
```

刪除物件

AWS ElementalMediaStore提供了從容器中刪除物件的不同選項：

- 刪除個別物件。無需支付費用。
- 清空容器以一次刪除容器內的所有物件。因為此程序使用 API 呼叫，所以會收取一般 API 費用。
- 新增物件生命週期政策，以便在物件達到特定存留期時刪除物件。無需支付費用。

刪除物件

您可以使用主控台或 AWS CLI 來個別刪除物件。或者，您可以新增物件生命週期政策，在物件達到容器中的特定存留期後自動刪除物件，或清空容器以刪除該容器內的所有物件。

> **Note**
>
> 當您刪除資料夾中的唯一物件時，AWS ElementalMediaStore會自動刪除此資料夾以及任何在該資料夾上層的空資料夾。例如，假設您有一個不含任何檔案、名為 `premium` 的資料夾，但其中包含一個名為 `canada` 的子資料夾。`canada` 子資料夾包含一個名為 `mlaw.ts` 的檔案。如果您刪除檔案 `mlaw.ts`，則服務會同時刪除 `premium` 和 `canada` 資料夾。

若要刪除物件 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，選擇具備您要刪除之物件的容器名稱。
3. 如果您想要刪除的物件是在某個資料夾中，請繼續選擇資料夾名稱，直到您看到該物件為止。
4. 選擇物件名稱左側的選項。
5. 選擇 Delete (刪除)。

若要刪除物件 (AWS CLI)

- 在 AWS CLI 中，使用 `delete-object` 命令。

  範例：

  ```
  aws mediastore-data --region us-west-2 delete-object --endpoint=https://
  aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com --path=/folder_name/
  README.md
  ```

  此命令沒有傳回值。

清空容器

您可以清空容器，以刪除儲存在容器內的所有物件。或者，您可以新增物件生命週期政策，在物件於容器中存在超過特定存留期後自動刪除物件，或者您可以個別刪除物件。

清空容器 (主控台)

1. 開啟MediaStore主控台https://console.aws.amazon.com/mediastore/。
2. 在 Containers (容器) 頁面上，為您要清空的容器選擇選項。

3. 選擇 Empty container (清空容器)。會出現確認訊息。
4. 在文字欄位中輸入容器名稱以確認您要清空的容器，然後選擇空白。

AWS Elemental 中的安全性 MediaStore

的雲端安全 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構是為了滿足最安全敏感組織的需求而建置。

安全性是 AWS 和 之間的共同責任。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 中執行 AWS 服務的基礎設施 AWS 雲端。 AWS 也為您提供可安全使用的服務。第三方稽核人員會定期測試和驗證我們安全的有效性，做為AWS 合規計畫的一部分。若要了解適用於 AWS Elemental 的合規計劃 MediaStore，請參閱AWS 合規計劃範圍內的服務。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。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。

本文件可協助您了解如何在使用 時套用共同的責任模型 MediaStore。下列主題說明如何設定 MediaStore 以符合您的安全和合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 MediaStore 資源。

主題

- AWS Elemental 中的資料保護 MediaStore
- AWS Elemental 的 Identity and Access Management MediaStore
- 在 中記錄和監控 AWS Elemental MediaStore
- AWS Elemental 的合規驗證 MediaStore
- AWS Elemental 中的復原能力 MediaStore
- AWS Elemental 中的基礎設施安全 MediaStore
- 預防跨服務混淆代理人

AWS Elemental 中的資料保護 MediaStore

AWS 共同責任模型適用於 AWS Elemental 中的資料保護 MediaStore。如此模型所述， AWS 負責保護執行所有 的全域基礎設施 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。FAQ如需歐洲資料保護的相關資訊，請參閱AWS 安全部落格上的AWS 共同責任模型和GDPR部落格文章。

基於資料保護目的，我們建議您保護 AWS 帳戶 登入資料，並使用 AWS IAM Identity Center 或 AWS Identity and Access Management () 設定個別使用者IAM。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：

- 對每個帳戶使用多重驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 和 建議 TLS 1.3。
- 使用 設定 API 和使用者活動記錄 AWS CloudTrail。如需有關使用 CloudTrail 線索擷取 AWS 活動的資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用 CloudTrail 線索。
- 使用 AWS 加密解決方案，以及其中的所有預設安全控制 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 AWS 透過命令列界面或 FIPS 存取 時需要 140-3 個經過驗證的密碼編譯模組API，請使用FIPS端點。如需可用FIPS端點的詳細資訊，請參閱聯邦資訊處理標準 (FIPS) 140-3。

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 MediaStore 或其他 AWS 服務 使用 主控台API AWS CLI、 或 時 AWS SDKs。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。如果您將URL提供給外部伺服器，強烈建議您不要在 中包含登入資料資訊URL，以驗證您對該伺服器的請求。

資料加密

MediaStore 使用業界標準 AES-256 演算法加密靜態容器和物件。建議您使用 MediaStore 以下列方式保護資料：

- 建立容器政策，以控制該容器中所有資料夾和物件的存取權。如需詳細資訊，請參閱the section called “容器政策”。
- 建立跨來源資源共享 (CORS) 政策，以允許選擇性地存取您的 MediaStore 資源的跨來源。使用 CORS，您可以允許載入某個網域的用戶端 Web 應用程式與不同網域中的資源互動。如需詳細資訊，請參閱the section called “CORS 政策”。

AWS Elemental 的 Identity and Access Management MediaStore

AWS Identity and Access Management (IAM) 是 AWS 服務 ，可協助管理員安全地控制對 AWS 資源的存取。 IAM管理員會控制誰可以驗證 （ 登入 ） 和授權 （ 具有許可 ） 來使用 MediaStore 資源。 IAM 是您可以免費使用 AWS 服務 的 ，無需額外費用。

主題

目標對象

使用方式 AWS Identity and Access Management (IAM) 會有所不同，取決於您在其中執行的工作 MediaStore。

服務使用者 – 如果您使用 MediaStore 服務來執行您的任務，則您的管理員會為您提供所需的登入資料和許可。當您使用更多 MediaStore 功能來執行工作時，您可能需要額外的許可。了解存取的管理方式可協助您向管理員請求正確的許可。若您無法存取 MediaStore 中的某項功能，請參閱 對 AWS Elemental MediaStore 身分和存取進行故障診斷。

服務管理員 – 如果您負責公司 MediaStore 的資源，您可能擁有 的完整存取權 MediaStore。您的任務是判斷服務使用者應存取 MediaStore 的功能和資源。您接著必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，瞭解 IAM 的基本概念。若要進一步了解貴公司如何 IAM搭配 使用 MediaStore，請參閱 AWS Elemental 如何使用 MediaStore IAM。

IAM 管理員：如果您是IAM管理員，建議您了解如何撰寫政策以管理存取權的詳細資訊 MediaStore。若要檢視您可以在 中使用的以 MediaStore 身分為基礎的政策範例IAM，請參閱 AWS Elemental 的身分型政策範例 MediaStore。

使用身分驗證

身分驗證是您 AWS 使用身分憑證登入 的方式。您必須以 AWS 帳戶根使用者身分、 IAM使用者身分或擔任 IAM角色來驗證 （ 登入 AWS)。

您可以使用透過身分來源提供的憑證，以聯合身分 AWS 身分身分身分身分登入 。 AWS IAM Identity Center (IAM Identity Center) 使用者、您公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料，都是聯合身分的範例。當您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使用聯合 AWS 身分存取 時，您會間接擔任角色。

視您身分的使用者類型而定，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需登入的詳細資訊 AWS，請參閱AWS 登入 《 使用者指南》中的如何登入您的 AWS 帳戶 。

如果您以 AWS 程式設計方式存取 ， AWS 會提供軟體開發套件 (SDK) 和命令列界面 (CLI)，以使用您的 登入資料以密碼編譯方式簽署您的請求。如果您不使用 AWS 工具，則必須自行簽署請求。如需使用建議方法自行簽署請求的詳細資訊，請參閱IAM《 使用者指南》中的AWS API請求的簽章版本 4。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如， AWS 建議您使用多重驗證 (MFA) 來提高帳戶的安全性。若要進一步了解，請參閱AWS IAM Identity Center 《 使用者指南》中的多重要素驗證和》 IAM 使用者指南》AWS 中的多重要素驗證IAM。

AWS 帳戶 根使用者

當您建立 時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取 帳戶中的所有 AWS 服務 和資源。此身分稱為 AWS 帳戶 Theroot 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需需要您以根使用者身分登入的任務完整清單，請參閱IAM《 使用者指南》中的需要根使用者憑證的任務。

聯合身分

最佳實務是， 要求人類使用者，包括需要管理員存取權的使用者，使用臨時登入資料 AWS 服務 與身分提供者聯合來存取 。

聯合身分是來自您企業使用者目錄、Web 身分提供者、 AWS Directory Service、身分中心目錄或任何使用透過身分來源提供的憑證 AWS 服務 存取的使用者。當聯合身分存取時 AWS 帳戶，它們會擔任角色，而角色會提供臨時登入資料。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，或者您可以連接並同步到您自己的身分來源中的一組使用者和群組，以用於所有 AWS 帳戶 和 應用程式。如需 IAM Identity Center 的相關資訊，請參閱AWS IAM Identity Center 《 使用者指南》中的什麼是 IAM Identity Center？。

IAM 使用者和群組

IAM 使用者是 中具有單一人員或應用程式特定許可 AWS 帳戶 的身分。如果可能，我們建議依賴臨時憑證，而不是建立具有密碼和存取金鑰等長期憑證IAM的使用者。不過，如果您有特定的使用案例需要IAM使用者長期登入資料，建議您輪換存取金鑰。如需詳細資訊，請參閱IAM《 使用者指南》中的針對需要長期憑證的使用案例定期輪換存取金鑰。

IAM 群組是一種指定 IAM 使用者集合的實體。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有名為 的群組IAMAdmins，並授予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供臨時憑證。若要進一步了解，請參閱IAM《 使用者指南》中的IAM使用者使用案例。

IAM 角色

IAM 角色是 中具有特定許可 AWS 帳戶 的身分。它與 IAM 使用者相似，但是不會與特定人員建立關聯。若要暫時在 中擔任IAM角色 AWS Management Console，您可以從使用者切換至IAM角色（主控台）。您可以呼叫 或 AWS API AWS CLI 操作，或使用自訂 來擔任角色URL。如需使用角色方法的詳細資訊，請參閱IAM《 使用者指南》中的擔任角色的方法。

使用臨時登入資料的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 — 如需向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需聯合角色的相關資訊，請參閱IAM《 使用者指南》中的為第三方身分提供者（聯合）建立角色。如果您使用 IAM Identity Center，您可以設定許可集。若要控制身分在驗證後可以存取哪些內容， IAM Identity Center 會將許可集與 中的角色相關聯IAM。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的許可集。
- 臨時IAM使用者許可 – IAM使用者或角色可以擔任 IAM 角色，暫時接受特定任務的不同許可。
- 跨帳戶存取權 – 您可以使用 IAM 角色，允許不同帳戶中的某人 (受信任的主體) 存取您帳戶的資源。角色是授予跨帳戶存取權的主要方式。不過，對於某些 AWS 服務，您可以直接將政策連接到資源（而不是使用角色做為代理）。若要了解跨帳戶存取的角色和資源型政策之間的差異，請參閱IAM《 使用者指南》中的跨帳戶資源存取IAM。
- 跨服務存取 – 有些 AWS 服務 使用其他 中的功能 AWS 服務。例如，當您在服務中呼叫 時，該服務通常會在 Amazon 中執行應用程式，EC2或在 Amazon S3 中存放物件。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
  - 轉送存取工作階段 (FAS) – 當您使用 IAM使用者或角色在其中執行動作時 AWS，您會被視為委託人。當您使用某些服務時，您可以執行 動作，然後在不同的服務中啟動另一個動作。 FAS會使用呼叫 的委託人許可 AWS 服務，並結合 對下游服務 AWS 服務 提出請求的請求。FAS只有在服務收到請求時，才會提出請求，要求 與其他 AWS 服務 或 資源互動以完成。在此情況下，您必須具有執行這兩個動作的許可。如需提出FAS請求時的政策詳細資訊，請參閱轉送存取工作階段。

- 服務角色 – 服務角色是IAM服務擔任的角色，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，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。
- 服務連結角色 – 服務連結角色是連結至 的服務角色類型 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

- 在 Amazon 上執行的應用程式 EC2 – 您可以使用 IAM角色來管理在EC2執行個體上執行之應用程式的臨時登入資料，以及提出 AWS CLI 或 AWS API 請求。這是在 EC2 執行個體內存放存取金鑰的較好方式。若要將 AWS 角色指派給EC2執行個體並將其提供給其所有應用程式，您可以建立連接至執行個體的執行個體描述檔。執行個體描述檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得暫時登入資料。如需詳細資訊，請參閱IAM《 使用者指南》中的使用 IAM角色將許可授予在Amazon EC2執行個體上執行的應用程式。

使用政策管理存取權

您可以透過建立政策並將其連接至身分或資源 AWS 來控制 AWS 中的存取。政策是 中的物件， AWS 當與身分或資源建立關聯時， 會定義其許可。當委託人 （使用者、根使用者或角色工作階段） 發出請求時， 會 AWS 評估這些政策。政策中的許可決定是否允許或拒絕請求。大多數政策會以JSON文件 AWS 形式存放在 中。如需JSON政策文件結構和內容的詳細資訊，請參閱IAM《 使用者指南》中的JSON政策概觀。

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。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。

預設情況下，使用者和角色沒有許可。若要授予使用者對所需資源執行動作的許可，IAM管理員可以建立IAM政策。然後，管理員可以將IAM政策新增至角色，使用者可以擔任角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該政策的使用者可以從 AWS Management Console、 AWS CLI或 AWS 取得角色資訊API。

身分型政策

身分型政策是您可以連接到身分的JSON許可政策文件，例如IAM使用者、使用者群組或角色。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分型政策，請參閱IAM《 使用者指南》中的使用客戶受管政策定義自訂IAM許可。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管政策是獨立的政策，您可以連接到 中的多個使用者、群組和角色 AWS 帳戶。受管政策包括 AWS 受管政策和客戶受管政策。若要了解如何在受管政策或內嵌政策之間進行選擇，請參閱IAM《 使用者指南》中的在受管政策和內嵌政策之間進行選擇。

資源型政策

以資源為基礎的政策是您連接到資源JSON的政策文件。資源型政策的範例包括IAM角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。主體可以包含帳戶、使用者、角色、聯合身分使用者，或 AWS 服務。

資源型政策是位於該服務中的內嵌政策。您無法在資源型政策IAM中使用來自 的 AWS 受管政策。

存取控制清單 (ACLs)

存取控制清單 (ACLs) 控制哪些主體 （帳戶成員、使用者或角色） 具有存取資源的許可。 ACLs 類似於以資源為基礎的政策，雖然它們不使用JSON政策文件格式。

Amazon S3 AWS WAF和 Amazon VPC是支援 的服務範例ACLs。若要進一步了解 ACLs，請參閱《Amazon Simple Storage Service 開發人員指南》中的存取控制清單 (ACL) 概觀。

其他政策類型

AWS 支援其他較不常見的政策類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可界限是一種進階功能，您可以在其中設定身分型政策可授予IAM實體 (IAM 使用者或角色） 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 `Principal` 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可界限的詳細資訊，請參閱IAM《 使用者指南》中的IAM實體的許可界限。
- 服務控制政策 (SCPs) – SCPs 是為組織或組織單位 (OU) in. 指定最大許可JSON的政策 AWS Organizations。 AWS Organizations 是一種用於分組和集中管理您企業擁有 AWS 帳戶 的多個的服務。如果您啟用組織中的所有功能，則可以將服務控制政策 (SCPs) 套用至任何或所有帳戶。SCP 限制成員帳戶中實體的許可，包括每個實體 AWS 帳戶根使用者。如需 Organizations 和 的詳細資訊 SCPs，請參閱AWS Organizations 《 使用者指南》中的服務控制政策。
- 資源控制政策 (RCPs) – RCPs 是可用來設定帳戶中資源最大可用許可JSON的政策，而無須更新連接至您擁有之每個資源IAM的政策。會RCP限制成員帳戶中資源的許可，並可能影響身分的有效

許可，包括 AWS 帳戶根使用者，無論它們是否屬於您的組織。如需 Organizations 和 的詳細資訊 RCPs，包括 AWS 服務 支援 的清單RCPs，請參閱AWS Organizations 《 使用者指南》中的資源控制政策 (RCPs)。

- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過撰寫程式的方式建立角色或聯合使用者的暫時工作階段時，做為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需詳細資訊，請參閱 IAM 使用者指南中的工作階段政策。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。若要了解如何 AWS 在涉及多種政策類型時決定是否允許請求，請參閱 IAM 使用者指南中的政策評估邏輯。

AWS Elemental 如何使用 MediaStore IAM

在您使用 IAM 管理 的存取權之前 MediaStore，請先了解哪些IAM功能可與 搭配使用 MediaStore。

IAM 您可以搭配 AWS Elemental 使用的功能 MediaStore

IAM 功能	MediaStore 支援
身分型政策	是
資源型政策	是
政策動作	是
政策資源	是
政策條件索引鍵 (服務特定)	是
ACLs	否
ABAC （政策中的標籤）	部分
臨時憑證	是
主體許可	是

IAM 功能	MediaStore 支援
服務角色	是
服務連結角色	否

若要深入了解 MediaStore 和其他 AWS 服務如何搭配大多數IAM功能使用，請參閱IAM《 使用者指南》中的 AWS 服務IAM。

的身分型政策 MediaStore

支援身分型政策：是

身分型政策是您可以連接到身分的JSON許可政策文件，例如IAM使用者、使用者群組或角色。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分型政策，請參閱IAM《 使用者指南》中的使用客戶受管政策定義自訂IAM許可。

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。您無法在身分型政策中指定主體，因為這會套用至連接的使用者或角色。若要了解您可以在JSON政策中使用的所有元素，請參閱IAM《 使用者指南》中的IAMJSON政策元素參考。

的身分型政策範例 MediaStore

若要檢視 MediaStore 身分型政策的範例，請參閱 AWS Elemental 的身分型政策範例 MediaStore。

中的資源型政策 MediaStore

支援資源型政策：是

以資源為基礎的政策是您連接到資源JSON的政策文件。資源型政策的範例包括IAM角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。主體可以包含帳戶、使用者、角色、聯合身分使用者，或 AWS 服務。

若要啟用跨帳戶存取，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，做為以資源為基礎政策的委託人。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當主體和資源位於不同的位置時 AWS 帳戶，信任帳戶中的IAM管理員也必須授予主體實體（使用者或角色）存取資源的許可。其透過

將身分型政策連接到實體來授與許可。不過，如果資源型政策會為相同帳戶中的主體授予存取，這時就不需要額外的身分型政策。如需詳細資訊，請參閱IAM《 使用者指南》中的跨帳戶資源存取IAM。

> **Note**
>
> MediaStore 也支援容器政策，以定義哪些主體實體（帳戶、使用者、角色和聯合身分使用者）可以在容器上執行動作。如需詳細資訊，請參閱容器政策。

的政策動作 MediaStore

支援政策動作：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。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。

JSON 政策的 `Action`元素說明您可以用來允許或拒絕政策中存取的動作。政策動作通常具有與相關聯 AWS API操作相同的名稱。有一些例外狀況，例如沒有相符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。

若要查看 MediaStore 動作清單，請參閱服務授權參考中的 AWS Elemental 定義的動作 MediaStore。

中的政策動作在動作之前 MediaStore 使用下列字首：

```
mediastore
```

若要在單一陳述式中指定多個動作，請用逗號分隔。

```
"Action": [
      "mediastore:action1",
      "mediastore:action2"
         ]
```

若要檢視 MediaStore 身分型政策的範例，請參閱 AWS Elemental 的身分型政策範例 MediaStore。

的政策資源 MediaStore

支援政策資源：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。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 政策元素會指定 動作套用的物件。陳述式必須包含 Resource 或 NotResource 元素。最佳實務是使用其 Amazon Resource Name (ARN) 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": "*"
```

若要查看 MediaStore 資源類型及其 的清單ARNs，請參閱服務授權參考中的 AWS Elemental 定義的資源 MediaStore。若要了解您可以使用哪些動作來指定每個資源ARN的 ，請參閱 AWS Elemental 定義的動作 MediaStore。

MediaStore 容器資源具有下列 ARN：

```
arn:${Partition}:mediastore:${Region}:${Account}:container/${containerName}
```

如需 格式的詳細資訊ARNs，請參閱 Amazon Resource Names (ARNs) AWS 和服務命名空間。

例如，若要在陳述式中指定AwardsShow容器，請使用下列 ARN：

```
"Resource": "arn:aws:mediastore:us-east-1:111122223333:container/AwardsShow"
```

的政策條件索引鍵 MediaStore

支援服務特定政策條件金鑰：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。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， 會使用邏輯OR操作 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。如需詳細資訊，請參閱IAM《 使用者指南》中的IAM政策元素：變數和標籤。

AWS 支援全域條件索引鍵和服務特定條件索引鍵。若要查看所有 AWS 全域條件索引鍵，請參閱IAM《 使用者指南》中的AWS 全域條件內容索引鍵。

若要查看 MediaStore 條件索引鍵清單，請參閱服務授權參考中的 AWS Elemental 的條件索引鍵 MediaStore。若要了解您可以使用條件索引鍵的動作和資源，請參閱 AWS Elemental 定義的動作 MediaStore。

若要檢視 MediaStore 身分型政策的範例，請參閱 AWS Elemental 的身分型政策範例 MediaStore。

ACLs 在 中 MediaStore

支援 ACLs：否

存取控制清單 (ACLs) 控制哪些主體（帳戶成員、使用者或角色）具有存取資源的許可。 ACLs 類似於以資源為基礎的政策，雖然它們不使用JSON政策文件格式。

ABAC 使用 MediaStore

支援 ABAC（政策中的標籤）：部分

屬性型存取控制 (ABAC) 是一種根據屬性定義許可的授權策略。在 中 AWS，這些屬性稱為標籤。您可以將標籤連接到IAM實體（使用者或角色）和許多 AWS 資源。標記實體和資源是 的第一步ABAC。然後，您可以設計ABAC政策，在主體的標籤符合其嘗試存取之資源上的標籤時允許操作。

ABAC 有助於快速成長的環境，並有助於處理政策管理變得繁瑣的情況。

如需根據標籤控制存取，請使用 aws:ResourceTag/*key-name*、aws:RequestTag/*key-name* 或 aws:TagKeys 條件索引鍵，在政策的條件元素中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 的詳細資訊ABAC，請參閱IAM《 使用者指南》中的使用ABAC授權定義許可。若要檢視包含設定之步驟的教學課程ABAC，請參閱IAM《 使用者指南》中的使用屬性型存取控制 (ABAC)。

搭配 使用臨時登入資料 MediaStore

支援臨時憑證：是

當您使用臨時憑證登入時，有些 AWS 服務 無法使用。如需詳細資訊，包括 AWS 服務 使用哪些臨時登入資料，請參閱IAM《 使用者指南》中的AWS 服務 使用 IAM 。

如果您 AWS Management Console 使用使用者名稱和密碼以外的任何方法登入 ，則表示您使用的是臨時登入資料。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取 時，該程序會自動建立臨時登入資料。當您以使用者身分登入主控台，然後切換角色時，也會自動建立臨時憑證。如需切換角色的詳細資訊，請參閱IAM《 使用者指南》中的從使用者切換到IAM角色 （ 主控台 ）。

您可以使用 AWS CLI 或 手動建立臨時登入資料 AWS API。然後，您可以使用這些臨時登入資料來存取 AWS。 AWS 建議您動態產生臨時登入資料，而不是使用長期存取金鑰。如需詳細資訊，請參閱 中的臨時安全登入IAM資料。

的跨服務主體許可 MediaStore

支援轉送存取工作階段 (FAS)：是

當您使用 IAM使用者或角色在 中執行動作時 AWS，您會被視為委託人。當您使用某些服務時，您可以執行 動作，然後在不同的服務中啟動另一個動作。 FAS會使用呼叫 的委託人許可 AWS 服務，並結合 對下游服務 AWS 服務 提出請求的請求。FAS只有在服務收到請求時，才會提出請求，要求 與其他 AWS 服務 或 資源互動以完成。在此情況下，您必須具有執行這兩個動作的許可。如需提出FAS請求時的政策詳細資訊，請參閱轉送存取工作階段。

MediaStore 的服務角色

支援服務角色：是

服務角色是IAM服務擔任的角色，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，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。

> **Warning**
>
> 變更服務角色的許可可能會中斷 MediaStore 功能。只有在 MediaStore 提供指引時，才能編輯服務角色。

的服務連結角色 MediaStore

支援服務連結角色：否

服務連結角色是連結至 的服務角色類型 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需建立或管理服務連結角色的詳細資訊，請參閱AWS 使用 的服務IAM。在表格中尋找服務，其中包含服務連結角色欄中的 Yes。選擇是連結，以檢視該服務的服務連結角色文件。

AWS Elemental 的身分型政策範例 MediaStore

根據預設，使用者和角色不具備建立或修改 MediaStore 資源的權限。他們也無法使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 來執行任務 AWS API。若要授予使用者對所需資源執行動作的許可，IAM管理員可以建立IAM政策。然後，管理員可以將IAM政策新增至角色，使用者可以擔任角色。

若要了解如何使用這些範例政策文件來建立以IAM身分為基礎的JSON政策，請參閱IAM《 使用者指南》中的建立IAM政策 （ 主控台 ）。

如需 定義的動作和資源類型的詳細資訊 MediaStore，包括ARNs每個資源類型的 格式，請參閱服務授權參考中的 AWS Elemental 的動作、資源和條件索引鍵 MediaStore。

主題

- 政策最佳實務
- 使用 MediaStore 主控台
- 允許使用者檢視他們自己的許可

政策最佳實務

以身分為基礎的政策會判斷是否有人可以建立、存取或刪除您帳戶中 MediaStore 的資源。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 受管政策並邁向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用 AWS 受管政策，將許可授予許多常見使用案例。它們可在您的 中使用 AWS 帳戶。我們建議您定義特定於使用案例 AWS 的客戶受管政策，以進一步減少許可。如需詳細資訊，請參閱IAM《 使用者指南》中的 AWS 受管政策或 AWS 任務函數的受管政策。
- 套用最低權限許可 – 當您使用 IAM 政策設定許可時， 只會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 套用許可的詳細資訊，請參閱IAM《 使用者指南》中的政策和許可IAM。

- 使用IAM政策中的條件來進一步限制存取：您可以將條件新增至政策，以限制對動作和資源的存取。例如，您可以撰寫政策條件來指定所有請求都必須使用 傳送SSL。如果透過特定 使用服務動作，您也可以使用條件來授予存取服務動作的權限 AWS 服務，例如 AWS CloudFormation。如需詳細資訊，請參閱IAM《 使用者指南》中的IAMJSON政策元素：條件。
- 使用 IAM Access Analyzer 驗證您的IAM政策，以確保安全且功能正常的許可 – IAM Access Analyzer 會驗證新的和現有的政策，使政策符合IAM政策語言 (JSON) 和IAM最佳實務。 IAMAccess Analyzer 提供超過 100 個政策檢查和可行的建議，以協助您撰寫安全且實用的政策。如需詳細資訊，請參閱IAM《 使用者指南》中的使用 IAM Access Analyzer 驗證政策。
- 需要多重要素驗證 (MFA) – 如果您的案例需要 IAM使用者或 中的根使用者 AWS 帳戶，請開啟 MFA 以獲得額外的安全性。若要在呼叫API操作MFA時要求 ，請將MFA條件新增至您的政策。如需詳細資訊，請參閱IAM《 使用者指南》中的使用 安全API存取MFA。

如需 中最佳實務的詳細資訊IAM，請參閱IAM《 使用者指南》中的安全最佳實務IAM。

使用 MediaStore 主控台

若要存取 AWS Elemental MediaStore 主控台，您必須擁有一組最低許可。這些許可必須允許您列出和檢視 中 MediaStore 資源的詳細資訊 AWS 帳戶。如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (使用者或角色) 而言，主控台就無法如預期運作。

對於僅對 AWS CLI 或 進行呼叫的使用者，您不需要允許最低主控台許可 AWS API。相反地，只允許存取與其嘗試執行API的操作相符的動作。

為了確保使用者和角色仍然可以使用 MediaStore 主控台，也請將 MediaStore *ConsoleAccess*或 *ReadOnly* AWS 受管政策連接到實體。如需詳細資訊，請參閱IAM《 使用者指南》中的新增許可給使用者。

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 使用者檢視連接到他們使用者身分的內嵌及受管政策。此政策包含在主控台上完成此動作的許可，或使用 AWS CLI 或 以程式設計方式完成此動作的許可 AWS API。

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

對 AWS Elemental MediaStore 身分和存取進行故障診斷

使用下列資訊來協助您診斷和修正使用 MediaStore 和 時可能遇到的常見問題IAM。

主題

- 我無權在 中執行動作 MediaStore
- 我無權執行 iam：PassRole
- 我想要允許 以外的人員 AWS 帳戶 存取我的 MediaStore 資源

我無權在 中執行動作 MediaStore

如果您收到錯誤，告知您未獲授權執行動作，您的政策必須更新，允許您執行動作。

當mateojacksonIAM使用者嘗試使用主控台檢視虛構*my-example-widget*資源的詳細資訊，但沒有虛構mediastore:*GetWidget*許可時，會發生下列範例錯誤。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 mediastore:GetWidget on resource: my-example-widget
```

在此情況下，必須更新 mateojackson 使用者的政策，允許使用 mediastore:*GetWidget* 動作存取 *my-example-widget* 資源。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我無權執行 iam：PassRole

如果您收到錯誤，告知您未獲授權執行 iam:PassRole 動作，您的政策必須更新，允許您將角色傳遞給 MediaStore。

有些 AWS 服務 可讓您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

以下範例錯誤會在名為 marymajor 的 IAM 使用者嘗試使用主控台在 MediaStore 中執行動作時發生。但是，動作請求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在這種情況下，Mary 的政策必須更新，允許她執行 iam:PassRole 動作。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我想要允許 以外的人員 AWS 帳戶 存取我的 MediaStore 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。對於支援資源型政策或存取控制清單 (ACLs) 的服務，您可以使用這些政策來授予人員存取您的資源。

如需進一步了解，請參閱以下內容：

- 若要了解 是否 MediaStore 支援這些功能，請參閱 AWS Elemental 如何使用 MediaStore IAM。
- 若要了解如何在您 AWS 帳戶 擁有的 資源間提供存取權，請參閱《 IAM 使用者指南》中的AWS 帳戶 在您擁有的另一個資源中提供IAM使用者存取權。

- 若要了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱IAM《 使用者指南》中的提供存取權給第三方 AWS 帳戶 擁有。
- 若要了解如何透過聯合身分提供存取權，請參閱IAM《 使用者指南》中的提供存取權給外部驗證的使用者 （ 聯合身分 ）。
- 若要了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱IAM《 使用者指南》中的跨帳戶資源存取IAM。

在 中記錄和監控 AWS Elemental MediaStore

本章節會概述 AWS Elemental MediaStore 中的記錄和監控選項，可讓您用來保障安全。如需 MediaStore 記錄和監控的詳細資訊，請參閱AWS Elemental 中的監控和標記 MediaStore。

監控是維護 和 AWS 解決方案可靠性、可用性 AWS Elemental MediaStore 和效能的重要部分。您應該從 AWS 解決方案的所有部分收集監控資料，以便在發生多點故障時更輕鬆地偵錯。 AWS 提供數種工具來監控您的 MediaStore 資源並回應潛在事件。

Amazon CloudWatch 警示

使用 CloudWatch 警示，您會在指定的期間內觀看單一指標。如果指標超過指定的閾值，通知會傳送至 Amazon SNS主題或 AWS Auto Scaling 政策。 CloudWatch alarms 不會因為動作處於特定狀態而叫用動作。必須是狀態已變更並維持了所指定的時間長度，才會呼叫動作。如需詳細資訊，請參閱使用 監控 CloudWatch。

AWS CloudTrail 日誌

CloudTrail 提供 AWS 中使用者、角色或服務所採取動作的記錄 AWS Elemental MediaStore。使用 所收集的資訊 CloudTrail，您可以判斷提出的請求 MediaStore、提出請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。如需詳細資訊，請參閱使用 記錄API通話 CloudTrail。

AWS Trusted Advisor

Trusted Advisor 利用從服務數十萬 AWS 客戶中學到的最佳實務。 會 Trusted Advisor 檢查您的AWS 環境，然後在有機會節省成本、改善系統可用性和效能，或協助解決安全漏洞時提出建議。所有 AWS 客戶都可以存取五個 Trusted Advisor 檢查。擁有商業或企業支援計劃的客戶可以檢視所有 Trusted Advisor 檢查。

如需詳細資訊，請參閱AWS Trusted Advisor。

AWS Elemental 的合規驗證 MediaStore

若要了解 是否 AWS 服務 在特定合規計劃的範圍內，請參閱AWS 服務 合規計劃範圍內然後選擇您感興趣的合規計劃。如需一般資訊，請參閱 AWS Compliance Programs。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱在 中下載報告 AWS Artifact。

使用 時的合規責任 AWS 服務 取決於資料的敏感度、您公司的合規目標，以及適用的法律和法規。AWS 提供下列資源以協助合規：

- 安全合規與治理 - 這些解決方案實作指南內容討論了架構考量，並提供部署安全與合規功能的步驟。
- Amazon Web Services HIPAA上安全與合規架構 – 此白皮書說明公司如何使用 AWS 來建立HIPAA符合 資格的應用程式。

> **Note**
>
> 並非所有 AWS 服務 都HIPAA符合資格。如需詳細資訊，請參閱HIPAA合格服務參考。

- AWS 合規資源 – 此工作手冊和指南的集合可能適用於您的產業和位置。
- AWS 客戶合規指南 – 透過合規的角度了解共同的責任模型。本指南摘要說明保護 的最佳實務，AWS 服務 並將指南映射到跨多個架構的安全控制 （ 包括國家標準和技術研究所 ()NIST、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO))。
- AWS Config 開發人員指南中的使用規則評估資源 – AWS Config 服務會評估資源組態符合內部實務、產業準則和法規的程度。
- AWS Security Hub – 這 AWS 服務 可讓您全面檢視其中的安全狀態 AWS。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的法規遵循是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱 Security Hub controls reference。
- Amazon GuardDuty – 這會監控您的環境是否有可疑和惡意活動，藉此 AWS 服務 偵測對您 AWS 帳戶、工作負載、容器和資料的潛在威脅。 GuardDuty 可以透過滿足特定合規架構所強制要求的入侵偵測需求DSS，協助您解決各種合規要求，例如 PCI 。
- AWS Audit Manager – 這 AWS 服務 可協助您持續稽核 AWS 用量，以簡化您管理風險的方式，以及符合法規和產業標準的方式。

AWS Elemental 中的復原能力 MediaStore

AWS 全域基礎設施是以 AWS 區域 和 可用區域為基礎建置。 AWS 區域 提供多個實體分隔和隔離的可用區域，這些區域與低延遲、高輸送量和高備援聯網連接。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 AWS 區域 和 可用區域的詳細資訊，請參閱AWS 全域基礎設施。

除了 AWS 全球基礎設施之外， MediaStore 還提供多種功能，以協助支援您的資料彈性和備份需求。

AWS Elemental 中的基礎設施安全 MediaStore

AWS Elemental 是受管服務，受到 AWS 全球網路安全的 MediaStore 保護。如需 AWS 安全服務以及如何 AWS 保護基礎設施的相關資訊，請參閱AWS 雲端安全。若要使用基礎設施安全的最佳實務設計您的 AWS 環境，請參閱 Security Pillar AWS Well-Architected Framework 中的基礎設施保護。

您可以使用 AWS 已發佈的API呼叫， MediaStore 透過網路存取 。使用者端必須支援下列專案：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 和建議 TLS 1.3。
- 具有完美前向秘密 (PFS) 的加密套件，例如 DHE(Ephemeral Diffie-Hellman) 或 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)。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 委託人相關聯的私密存取金鑰來簽署。或者，您可以使用 AWS Security Token Service (AWS STS) 產生臨時安全憑證來簽署請求。

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代理人問題。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。

我們建議在資源政策中使用 `aws:SourceArn`和 `aws:SourceAccount`全域條件內容索引鍵，以限制 AWS Elemental 為資源 MediaStore 提供其他服務的許可。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`。

防範混淆代理人問題最有效的方式，是使用具有完整資源ARN的`aws:SourceArn`全域條件內容金鑰。如果您不知道資源ARN的完整內容，或如果您要指定多個資源，請針對 的未知部分使用具有萬用字元 (*) 的`aws:SourceArn`全域內容條件索引鍵ARN。例如：`arn:aws:servicename:*:123456789012:*`。

如果該aws:SourceArn值不包含帳戶 ID，例如 Amazon S3 儲存貯體 ARN，您必須使用兩個全域條件內容索引鍵來限制許可。

的值aws:SourceArn必須是在您的區域和帳戶中 MediaStore 發佈 CloudWatch 日誌的組態。

下列範例示範如何使用 中的 aws:SourceArn和 aws:SourceAccount全域條件內容索引鍵 MediaStore ，以防止混淆代理人問題。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "servicename.amazonaws.com"
    },
    "Action": "servicename:ActionName",
    "Resource": [
      "arn:aws:servicename:::ResourceName/*"
    ],
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:servicename:*:123456789012:*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

AWS Elemental 中的監控和標記 MediaStore

監控是維護 AWS Elemental 和其他 AWS 解決方案的可靠性、可用性 MediaStore 和效能的重要部分。 AWS 提供下列監控工具來監看 MediaStore、報告錯誤，並在適當時採取自動動作：

- AWS CloudTrail 會擷取您 AWS 帳戶或其代表發出的API呼叫和相關事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。您可以識別呼叫的使用者和帳戶 AWS、進行呼叫的來源 IP 地址，以及呼叫的時間。如需詳細資訊，請參閱《AWS CloudTrail 使用者指南》https://docs.aws.amazon.com/awscloudtrail/latest/userguide/。
- Amazon CloudWatch 會 AWS 即時監控您的 AWS 資源和您在 上執行的應用程式。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 Amazon EC2執行個體的CPU用量或其他指標，並在需要時自動啟動新的執行個體。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。
- Amazon CloudWatch Events 提供描述 AWS 資源變更的系統事件串流。一般而言， AWS 服務會在幾秒鐘內將事件通知交付至 CloudWatch 事件，但有時可能需要一分鐘或更長的時間。 CloudWatch 事件可啟用自動事件驅動運算，因為您可以撰寫規則來監控特定事件，並在發生這些事件 AWS 時觸發其他服務中的自動動作。如需詳細資訊，請參閱 Amazon CloudWatch Events 使用者指南。
- Amazon CloudWatch Logs 可讓您從 Amazon EC2執行個體和其他來源監控 CloudTrail、存放和存取您的日誌檔案。 CloudWatch Logs 可以監控日誌檔案中的資訊，並在達到特定閾值時通知您。您也可以將日誌資料存檔在高耐用性的儲存空間。如需詳細資訊，請參閱 Amazon CloudWatch Logs 使用者指南。

您也可以以標籤的形式將中繼資料指派給 MediaStore 容器。每個標籤都包含您定義的金鑰和值。標籤可讓您更輕鬆地管理、搜尋和篩選資源。您可以使用標籤在 AWS 管理主控台中組織 AWS 資源、建立所有資源的 AWS 用量和帳單報告，以及在基礎設施自動化活動期間篩選資源。

主題

使用 記錄AWS元素 MediaStore API呼叫 AWS CloudTrail

AWS Elemental MediaStore 已與 整合 AWS CloudTrail，此服務提供使用者、角色或服務 AWS in. CloudTrail captures 所採取動作的記錄 MediaStore。 會將 的API呼叫子集擷取 MediaStore 為事件，

包括來自主控台的 MediaStore呼叫，以及來自程式碼呼叫到 的呼叫 MediaStore API。如果您建立追蹤，您可以啟用 CloudTrail 事件持續交付至 Amazon S3 儲存貯體，包括 的事件 MediaStore。如果您未設定追蹤，仍然可以在 CloudTrail 主控台的事件歷史記錄中檢視最新的事件。使用 收集的資訊 CloudTrail，您可以判斷提出請求的請求 MediaStore、提出請求的 IP 地址、提出請求的人員、提出請求的時間等等。

若要進一步了解 CloudTrail，包括如何設定和啟用，請參閱 AWS CloudTrail 使用者指南。

主題

AWS 中的元素 MediaStore資訊 CloudTrail

CloudTrail 當您建立 AWS 帳戶時， 會在您的帳戶上啟用 。在 AWS Elemental 中發生支援的事件活動時 MediaStore，該活動會與事件歷史記錄中的 CloudTrail其他服務 AWS 事件一起記錄在事件中。您可以在 AWS 帳戶中檢視、搜尋和下載最近的事件。如需詳細資訊，請參閱使用事件歷史記錄檢視 CloudTrail 事件。

若要持續記錄您 AWS 帳戶中的事件，包括 的事件 MediaStore，請建立追蹤。線索可讓 CloudTrail 將日誌檔案交付至 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 AWS 區域。追蹤會記錄 AWS 分割區中所有 區域的事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析 CloudTrail 日誌中收集的事件資料並對其採取行動。如需詳細資訊，請參閱下列主題：

- 建立追蹤的概觀
- CloudTrail 支援的服務和整合
- 設定 的 Amazon SNS Notifications CloudTrail
- 從多個區域接收 CloudTrail 日誌檔案，以及從多個帳戶接收 CloudTrail 日誌檔案

AWS Elemental MediaStore 支援將下列操作記錄為 CloudTrail 日誌檔案中的事件：

- `CreateContainer`
- `DeleteContainer`
- `DeleteContainerPolicy`
- `DeleteCorsPolicy`

- DescribeContainer
- GetContainerPolicy
- GetCorsPolicy
- ListContainers
- PutContainerPolicy
- PutCorsPolicy

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：

- 是否使用根使用者或使用者登入資料提出請求
- 提出該請求時，是否使用了特定角色或聯合身分使用者的臨時安全憑證
- 請求是否由其他 AWS 服務提出

如需詳細資訊，請參閱 CloudTrail userIdentity 元素。

範例：AWS元素 MediaStore 日誌檔案項目

追蹤是一種組態，可讓您將事件以日誌檔案形式交付到您指定的 Amazon S3 儲存貯體。 CloudTrail 日誌檔案包含一或多個日誌項目。事件代表來自任何來源的單一請求，並包含所請求動作、動作的日期和時間、請求參數等相關資訊。 CloudTrail 日誌檔案不是公開API呼叫的排序堆疊追蹤，因此不會以任何特定順序顯示。

下列範例顯示示範 CreateContainer操作的 CloudTrail 日誌項目：

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "ABCDEFGHIJKL123456789",
        "arn": "arn:aws:iam::111122223333:user/testUser",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "testUser",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-07-09T12:55:42Z"
```

```
            }
        },
        "invokedBy": "signin.amazonaws.com"
    },
    "eventTime": "2018-07-09T12:56:54Z",
    "eventSource": "mediastore.amazonaws.com",
    "eventName": "CreateContainer",
    "awsRegion": "ap-northeast-1",
    "sourceIPAddress": "54.239.119.16",
    "userAgent": "signin.amazonaws.com",
    "requestParameters": {
        "containerName": "TestContainer"
    },
    "responseElements": {
        "container": {
            "status": "CREATING",
            "creationTime": "Jul 9, 2018 12:56:54 PM",
            "name": " TestContainer ",
            "aRN": "arn:aws:mediastore:ap-northeast-1:111122223333:container/
TestContainer"
        }
    },
    "requestID":
 "MNCTGH4HRQJ27GRMBVDPIVHEP4LO2BN6MUVHBCPSHOAWNSOKSXCO24B2UEOBBND5DONRXTMFK3TOJ4G7AHWMESI",
    "eventID": "7085b140-fb2c-409b-a329-f567912d704c",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


MediaStore 使用 Amazon 監控 AWS Elemental CloudWatch

您可以使用 監控 AWS Elemental MediaStore CloudWatch，這會收集原始資料並將其處理為可讀指標。 會 CloudWatch 保留統計資料 15 個月，以便您可以存取歷史資訊，並更清楚地了解 Web 應用程式或服務的效能。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。

AWS 提供下列監控工具，供您監看 MediaStore、回報錯誤，並適時採取自動動作：

- Amazon CloudWatch Logs 可讓您從 AWS AWS Elemental 等服務監控、存放和存取您的日誌檔案 MediaStore。您可以使用 CloudWatch Logs 來監控使用日誌資料的應用程式和系統。例如，CloudWatch Logs 可以追蹤應用程式日誌中發生的錯誤數量，並在錯誤率超過您指定的閾值時傳送通知給您。 CloudWatch Logs 使用您的日誌資料進行監控，因此不需要變更程式碼。例如，您可

以監控應用程式日誌中的特定文字詞彙（例如 "ValidationException"），或計算在特定期間內提出的`PutObject`請求數量。找到您要搜尋的詞彙時， CloudWatch Logs CloudWatch 會將資料報告至您指定的指標。日誌資料會在移轉和靜態時加密。

- Amazon CloudWatch Events 提供描述 AWS 資源變更的系統事件，例如 MediaStore 物件。一般而言， AWS 服務會在幾秒鐘內將事件通知傳遞至 CloudWatch 事件，但有時可能需要一分鐘或更長的時間。您可以設定規則來比對事件（例如`DeleteObject`請求），並將它們路由到一或多個目標函數或串流。 CloudWatch 事件會在發生操作變更時注意到它們。此外， CloudWatch Events 會回應這些操作變更，並視需要採取修正動作，透過傳送訊息來回應環境、啟用 函數、進行變更和擷取狀態資訊。

CloudWatch 日誌

存取記錄會為對容器中的物件所做的請求提供詳細的記錄。存取記錄對於許多應用程式都相當實用，例如安全和存取稽核。他們也可以協助您了解您的客戶群，並了解您的 MediaStore 帳單。 CloudWatch 日誌分類如下：

- 日誌串流是共享相同來源的一系列日誌事件。
- 日誌群組是共享相同保留、監控和存取控制設定的日誌串流群組。當您在容器上啟用存取記錄時，會 MediaStore 建立名稱為 的日誌群組`/aws/mediastore/MyContainerName`。您可以定義日誌群組，並指定放入每個群組的串流。可以屬於一個日誌群組的日誌串流數量並沒有配額。

根據預設，日誌將無限期保留且永遠不會過期。您可以調整每個日誌群組的保留政策，維持無限期保留，或選擇 1 天至 10 年之間的保留期間。

設定 Amazon 的許可 CloudWatch

使用 AWS Identity and Access Management (IAM) 建立角色，讓 AWS Elemental MediaStore 存取 Amazon CloudWatch。您必須執行這些步驟，才能為您的帳戶發佈 CloudWatch 日誌。 CloudWatch 會自動發佈您帳戶的指標。

允許 MediaStore 存取 CloudWatch

1. 開啟位於 IAM 的 https://console.aws.amazon.com/iam/ 主控台。
2. 在 IAM 主控台的導覽窗格中，選擇 Policies (政策)，然後選擇 Create Policy (建立政策)。
3. 選擇 JSON索引標籤並貼上下列政策：

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:CreateLogGroup"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/mediastore/*"
        }
    ]
}
```

此政策允許 為您 AWS 帳戶中任何區域中的任何容器 MediaStore 建立日誌群組和日誌串流。

4. 選擇檢閱政策。
5. 在 Review policy (檢閱政策) 頁面上，針對 Name (名稱)，輸入 **MediaStoreAccessLogsPolicy**，然後選擇 Create policy (建立政策)。
6. 在 IAM 主控台的導覽窗格中，選擇 Roles (角色)，然後選擇 Create role (建立角色)。
7. 選擇 Another AWS account (另一個 AWS 帳戶) 角色類型。
8. 針對帳戶 ID，輸入 AWS 您的帳戶 ID。
9. 選擇下一步：許可。
10. 在搜尋方塊中，輸入 **MediaStoreAccessLogsPolicy**。
11. 選取新政策旁的核取方塊，然後選擇 Next: Tags (下一步：標籤)。
12. 選擇 Next: Review (下一步：檢閱) 來預覽新使用者。
13. 針對角色名稱，輸入 **MediaStoreAccessLogs**，然後選擇建立角色。
14. 在確認訊息中，選擇您剛建立的角色名稱 (**MediaStoreAccessLogs**)。
15. 在角色的 Summary (摘要) 頁面上，選擇 Trust relationships (信任關係) 標籤。
16. 選擇編輯信任關係。

17. 在政策文件中，將委託人變更為 MediaStore 服務。它應該如下所示：

```
"Principal": {
   "Service": "mediastore.amazonaws.com"
},
```

整個政策看起來應該如下所示：

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

18. 選擇 Update Trust Policy (更新信任政策)。

啟用容器的存取記錄

根據預設， AWS Elemental MediaStore 不會收集存取日誌。當您在容器上啟用存取日誌時， 會將該容器中存放物件的存取日誌 MediaStore 傳遞給 Amazon CloudWatch。這些存取日誌會為對容器中存放的任何物件所做的請求提供詳細的記錄。這資訊可能包含要求類型、要求中指定的資源，以及要求的處理時間和日期。

> **Important**
>
> 在 MediaStore 容器上啟用存取記錄不會產生額外費用。不過，此服務提供給您的任何日誌檔都會產生一般儲存費用。(您可以隨時刪除這些日誌檔。) AWS 不會估算日誌檔傳遞的資料傳輸費，但會收取用於存取日誌檔之一般資料傳輸率的費用。

若要啟用存取記錄 (AWS CLI)

- 在 中 AWS CLI，使用 `start-access-logging`命令：

```
aws mediastore start-access-logging --container-name LiveEvents --region us-west-2
```

此命令沒有傳回值。

停用容器的存取記錄

當您在容器上停用存取記錄時，AWSElemental 會 MediaStore 停止傳送存取日誌至 Amazon CloudWatch。系統不會儲存這些存取日誌而且這些日誌不供擷取。

停用存取記錄 (AWS CLI)

- 在 中 AWS CLI，使用 `stop-access-logging`命令：

```
aws mediastore stop-access-logging --container-name LiveEvents --region us-west-2
```

此命令沒有傳回值。

對 AWS Elemental 中的存取日誌進行故障診斷 MediaStore

當 AWS Elemental MediaStore 存取日誌未出現在 Amazon 中時 CloudWatch，請參閱下表了解潛在原因和解決方案。

> **Note**
>
> 請務必啟用 AWS CloudTrail Logs 以協助進行疑難排解程序。

徵狀	問題可能是 ...	嘗試這麼做 ...
即使已啟用 CloudTrail 日誌，您也不會看到任何 CloudTrail 事件。	IAM 角色不存在，或其名稱、許可或信任政策不正確。	建立具有正確名稱、許可和信任政策的角色。請參閱 the section called “設定 的許可 CloudWatch”。

徵狀	問題可能是 ...	嘗試這麼做 ...
您已提交DescribeContainer API 請求，但回應顯示 AccessLoggingEnabled 參數的值為 False。此外，您沒有看見適用於進行成功 DescribeLogGroup 、CreateLogGroup 、DescribeLogStream 或 CreateLogStream 呼叫之 MediaStoreAccessLogs 角色 的任何 CloudTrail 事件。	IAM 角色不存在，或其名稱、許可或信任政策不正確。	建立具有正確名稱、許可和信任政策的角色。請參閱 the section called “設定 的許可 CloudWatch”。
	容器上沒有啟用存取記錄。	啟用容器的存取日誌。請參閱 the section called “啟用存取記錄”。

在 CloudTrail 主控台上，您會看到與MediaStoreAccessLogs 角色相關的存取遭拒錯誤的事件。 CloudTrail 事件可能包含如下的行：
 	
`"eventSource": "logs.amazonaws.com",`
 	
`"errorCode": "AccessDenied",`
 	
`"errorMessage": "User: arn:aws:sts::111122223333:assumed-role/MediaStoreAccessLogs/MediaStoreAccessLogsSession is not authorized to perform: logs:DescribeLogGroups on resource: arn:aws:logs:us-west-2:111122223333:log-group::log-stream:",`
	IAM 角色沒有 AWS Elemental 的正確許可 MediaStore。	更新IAM角色，以取得正確的許可和信任政策。請參閱 the section called “設定 的許可 CloudWatch”。

徵狀	問題可能是 ...	嘗試這麼做 ...
您沒有看到適用於一或多個整個容器的任何日誌。	您的帳戶可能超過每個區域每個帳戶日誌群組的 CloudWatch 配額。請參閱《Amazon CloudWatch Logs 使用者指南》中的日誌群組配額。	在 CloudWatch 主控台上，判斷您的帳戶是否符合日誌群組的 CloudWatch 配額。如果必要，請求增加配額。
您會看到一些登入， CloudWatch但並非所有預期看到的日誌。	您的帳戶可能已超過每個區域每個帳戶每秒交易的 CloudWatch 配額。請參閱《Amazon CloudWatch Logs 使用者指南PutLogEvents 》中的 配額。	請求提高每個區域每個帳戶每秒 CloudWatch 交易的配額。

存取日誌格式

存取日誌檔案包含一系列 JSON格式化日誌記錄，其中每個日誌記錄代表一個請求。日誌中欄位的順序可能有所不同。以下是包含兩個日誌記錄的範例日誌：

```
{
  "Path": "/FootballMatch/West",
  "Requester": "arn:aws:iam::111122223333:user/maria-garcia",
  "AWSAccountId": "111122223333",
  "RequestID":
 "aaaAAA111bbbBBB222cccCCC333dddDDD444eeeEEE555fffFFF666gggGGG777hhhHHH888iiiIII999jjjJJJ",
  "ContainerName": "LiveEvents",
  "TotalTime": 147,
  "BytesReceived": 1572864,
  "BytesSent": 184,
  "ReceivedTime": "2018-12-13T12:22:06.245Z",
  "Operation": "PutObject",
  "ErrorCode": null,
  "Source": "192.0.2.3",
```

```
  "HTTPStatus": 200,
  "TurnAroundTime': 7,
  "ExpiresAt": "2018-12-13T12:22:36Z"
}
 {
  "Path": "/FootballMatch/West",
  "Requester": "arn:aws:iam::111122223333:user/maria-garcia",
  "AWSAccountId": "111122223333",
  "RequestID":
 "dddDDD444eeeEEE555fffFFF666gggGGG777hhhHHH888iiiIII999jjjJJJ000cccCCC333bbbBBB222aaaAAA",
  "ContainerName": "LiveEvents",
  "TotalTime": 3,
  "BytesReceived": 641354,
  "BytesSent": 163,
  "ReceivedTime": "2018-12-13T12:22:51.779Z",
  "Operation": "PutObject",
  "ErrorCode": "ValidationException",
  "Source": "198.51.100.15",
  "HTTPStatus": 400,
  "TurnAroundTime": 1,
  "ExpiresAt": null
}
```

下列清單說明日誌記錄欄位：

AWSAccountId

用於提出請求的帳戶的帳戶 AWS ID。

BytesReceived

MediaStore 伺服器所接收到請求內文中的位元組數。

BytesSent

MediaStore 伺服器所傳送請求內文中的位元組數。此值通常與在伺服器回應中包含的 `Content-Length` 標頭值相同。

ContainerName

收到請求的容器名稱。

ErrorCode

MediaStore 錯誤碼（例如 `InternalServerError`)。如果沒有錯誤發生，則會顯示 - 字元。即使狀態碼為 200 (表示關閉連線或在伺服器開始串流回應後發生錯誤)，仍可能會顯示錯誤碼。

ExpiresAt

物件的過期日期和時間。此值是根據套用到容器的生命週期政策transient data rule中由 設定的過期時間。此值是 ISO-8601 日期時間，且會根據服務請求之主機的系統時鐘。如果生命週期政策沒有適用於物件的暫時性資料規則，或如果沒有套用生命週期政策到容器，此欄位的值為 null。此欄位僅適用於下列操作：PutObject、DescribeObject、 GetObject和 DeleteObject。

HTTPStatus

回應的數值HTTP狀態碼。

作業

已執行的操作，例如 PutObject 或 ListItems。

路徑

容器中的路徑，此容器為物件的存放位置。如果操作不會採用路徑參數，則會出現 - 字元。

ReceivedTime

收到請求的時間。此值是 ISO-8601 日期時間，且會根據服務請求之主機的系統時鐘。

要求者

用於提出請求之帳戶的使用者 Amazon Resource Name (ARN)。若是未經授權的請求，這個值會是 anonymous。如果在身分驗證完成前要求失敗，則此欄位可能會從日誌中遺失。對於這類要求，ErrorCode 可能會識別授權問題。

RequestID

AWS Elemental 產生的字串 MediaStore ，用於唯一識別每個請求。

來源

請求者或發出呼叫之服務的 AWS 服務主體的表觀網際網路地址。如果中間代理伺服器與防火牆會將提出請求的機器地址模糊處理，此值會設為 Null。

TotalTime

從伺服器角度計算的請求所經過的毫秒數 (ms)。此值是從服務收到您請求的時間開始，計算到回應傳送出最後一組位元組的時間。從用戶端角度進行的測量會受網路延遲影響，因此會從伺服器角度來進行此值的測量。

TurnAroundTime

處理您的請求所 MediaStore 花費的毫秒數。此值是從收到您要求的最後位元組的時間開始，計算到回應傳送出第一組位元組的時間。

日誌中欄位的順序可能有所不同。

記錄狀態變更會在一段時間後生效

記錄容器狀態的變更，要一段時間後才會實際影響到日誌檔交付。例如，若已啟用容器記錄，則在接下來的一小時內提出之請求，可能有些會記錄下來，有些則不會。如果您停用容器 B 的記錄，系統可能會繼續交付下一個小時的日誌，有些則不會。在所有情況下，新的設定最終都會生效，您無需採取任何進一步動作。

伺服器日誌交付最佳作法

存取日誌記錄會依最佳作法交付。已適合設定為在交付日誌記錄中記錄結果的容器的多數請求。大多數的日誌記錄會於記錄後的數小時內交付，但也可以常交付。

並不保證存取記錄的完成程度與時間先後順序。特定要求的日誌記錄，可能會在實際處理要求之後很久才交付，或者有可能完全不會交付。存取日誌的目的在於讓您能了解容器流量的真實狀態。雖然日誌記錄極少會缺失，但存取記錄並不代表所有請求的完整記錄。

其遵循存取記錄功能的最佳努力本質，AWS入口網站上可用的用量報告 (上的帳單和成本管理報告AWS Management Console) 可能包含一或多個未出現在已交付存取日誌中的存取請求。

存取記錄格式的程式設計考量

有時候，我們可能要在新增欄位，來擴展存取日誌的格式。必須寫入剖析存取日誌的程式碼，才能處理它不了解的其他欄位。

CloudWatch 活動

Amazon CloudWatch Events 可讓您自動化您的 AWS 服務，並自動回應系統事件，例如應用程式可用性問題或資源變更。您可編寫簡單的規則，來指示您在意的事件，以及當事件符合規則時所要自動執行的動作。

> **Important**
>
> 一般而言， AWS 服務會在幾秒鐘內將事件通知傳遞至 CloudWatch 事件，但有時可能需要一分鐘或更長的時間。

當檔案上傳到容器或從容器移除時，服務中會連續觸發兩個事件 CloudWatch ：

1. the section called “物件狀態變更事件”

2. the section called “容器狀態變更事件”

如需訂閱這些事件的資訊，請參閱 Amazon CloudWatch。

可以自動觸發的動作如下：

- 叫用 AWS Lambda 函數
- 叫用 Amazon EC2 Run 命令
- 將事件轉傳至 Amazon Kinesis Data Streams
- 啟用 AWS Step Functions 狀態機器
- 通知 Amazon SNS主題或 AWS SMS 佇列

搭配 AWS Elemental 使用 CloudWatch Events 的一些範例 MediaStore 包括：

- 每當建立容器時啟用 Lambda 函數
- 刪除物件時通知 Amazon SNS主題

如需詳細資訊，請參閱 Amazon CloudWatch Events 使用者指南。

主題

- AWS 元素 MediaStore 物件狀態變更事件
- AWS 元素 MediaStore 容器狀態變更事件

AWS 元素 MediaStore 物件狀態變更事件

物件的狀態已變更時 (當物件已上傳或刪除時)，將會發佈此事件。

> **Note**
>
> 因暫時性資料規則而過期的物件不會在 CloudWatch過期時發出事件。

如需訂閱此事件的資訊，請參閱 Amazon CloudWatch。

物件已更新

```
{
```

```
  "version": "1",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "MediaStore Object State Change",
  "source": "aws.mediastore",
  "account": "111122223333",
  "time": "2017-02-22T18:43:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:mediastore:us-east-1:111122223333:MondayMornings/Episode1/
Introduction.avi"
  ],
  "detail": {
    "ContainerName": "Movies",
    "Operation": "UPDATE",
    "Path":"TVShow/Episode1/Pilot.avi",
    "ObjectSize":123456,
    "URL": "https://a832p1qeaznlp9.files.mediastore-us-west-2.com/Movies/
MondayMornings/Episode1/Introduction.avi"
  }
}
```

物件已移除

```
{
  "version": "1",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "MediaStore Object State Change",
  "source": "aws.mediastore",
  "account": "111122223333",
  "time": "2017-02-22T18:43:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:mediastore:us-east-1:111122223333:Movies/MondayMornings/Episode1/
Introduction.avi"
  ],
  "detail": {
    "ContainerName": "Movies",
    "Operation": "REMOVE",
    "Path":"Movies/MondayMornings/Episode1/Introduction.avi",
    "URL": "https://a832p1qeaznlp9.files.mediastore-us-west-2.com/Movies/
MondayMornings/Episode1/Introduction.avi"
  }
}
```

AWS 元素 MediaStore 容器狀態變更事件

容器的狀態已變更時 (當容器已新增或刪除時)，將會發佈此事件。如需訂閱此事件的資訊，請參閱 Amazon CloudWatch。

容器已建立

```
{
  "version": "1",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "MediaStore Container State Change",
  "source": "aws.mediastore",
  "account": "111122223333",
  "time": "2017-02-22T18:43:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:mediastore:us-east-1:111122223333:container/Movies"
  ],
  "detail": {
    "ContainerName": "Movies",
    "Operation": "CREATE"
    "Endpoint": "https://a832p1qeaznlp9.mediastore-us-west-2.amazonaws.com"
  }
}
```

容器已移除

```
{
  "version": "1",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "MediaStore Container State Change",
  "source": "aws.mediastore",
  "account": "111122223333",
  "time": "2017-02-22T18:43:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:mediastore:us-east-1:111122223333:container/Movies"
  ],
  "detail": {
    "ContainerName": "Movies",
    "Operation": "REMOVE"
  }
```

```
}
```

MediaStore 使用 Amazon CloudWatch 指標監控 AWS Elemental

您可以使用 監控 AWS Elemental MediaStore CloudWatch，這會收集原始資料並將其處理為可讀指標。 CloudWatch 會保留統計資料 15 個月，以便您可以存取歷史資訊，並更清楚地了解 Web 應用程式或服務的效能。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。

對於 AWS Elemental MediaStore，您可能想要在指標達到特定閾值時，監看`BytesDownloaded`並傳送電子郵件給自己。

使用 CloudWatch 主控台檢視指標

指標會先依服務命名空間分組，再依各命名空間內不同的維度組合分類。

1. 登入 AWS Management Console 並在 開啟 CloudWatch 主控台https://console.aws.amazon.com/cloudwatch/。
2. 在導覽窗格中，選擇 指標。
3. 在所有指標下，選擇 AWS/MediaStore 命名空間。
4. 選擇要檢視指標的指標維度。例如，選擇 `Request metrics by container` 以檢視已傳送至容器之不同請求類型的指標。

使用 檢視指標 AWS CLI

- 在命令提示中，使用下列命令：

```
aws cloudwatch list-metrics --namespace "AWS/MediaStore"
```

AWS 元素 MediaStore 指標

下表列出 AWS Elemental MediaStore 傳送到的指標 CloudWatch。

> **Note**
>
> 若要檢視指標，您必須將指標政策新增至容器，以允許 將指標 MediaStore 傳送至 Amazon CloudWatch。

指標	描述
RequestCount	對容器提出的HTTP MediaStore請求總數，依操作類型 (Put、、GetDelete、Describe、) 分隔List。 單位：計數 有效維度： • 容器名稱 • 物件群組名稱 • 要求類型 有效的統計資訊：總和
4xxErrorCount	對 發出的HTTP請求數量 MediaStore ，導致 4xx 錯誤。 單位：計數 有效維度： • 容器名稱 • 物件群組名稱 • 要求類型 有效的統計資訊：總和
5xxErrorCount	對 發出的HTTP請求數量 MediaStore ，導致 5xx 錯誤。 單位：計數 有效維度： • 容器名稱 • 物件群組名稱 • 要求類型

指標	描述
	有效的統計資訊：總和
BytesUploaded	針對 MediaStore 容器提出之請求所上傳的位元組數目，且回應包含內文。 單位：位元組 有效維度： • 容器名稱 • 物件群組名稱 有效的統計資訊：平均 (每個請求的位元組數)、總和 (每個期間的位元組數)、範例計數、最小值 (相容於 P0.0)、最大值 (相同於 p100)，任何介於 p0.0 與 p99.9 的百分位數
BytesDownloaded	針對 MediaStore 容器提出之請求所下載的位元組數目，且回應包含內文。 單位：位元組 有效維度： • 容器名稱 • 物件群組名稱 有效的統計資訊：平均 (每個請求的位元組數)、總和 (每個期間的位元組數)、範例計數、最小值 (相容於 P0.0)、最大值 (相同於 p100)，任何介於 p0.0 與 p99.9 的百分位數

指標	描述
TotalTime	從伺服器角度計算的請求所經過的毫秒數。此值是從 MediaStore 接收請求的時間開始測量，直到傳送回應的最後一個位元組為止。從用戶端角度進行的測量會受網路延遲影響，因此會從伺服器角度來進行此值的測量。 單位：毫秒 有效維度： • 容器名稱 • 物件群組名稱 • 要求類型 有效的統計資訊：平均、最小值 (相容於 P0.0)、最大值 (相同於 p100)、任何介於 p0.0 與 p100 的百分位數
TurnaroundTime	處理您的請求所 MediaStore 花費的毫秒數。此值是從 MediaStore 接收請求最後一個位元組的時間，到傳送回應第一個位元組的時間進行測量。 單位：毫秒 有效維度： • 容器名稱 • 物件群組名稱 • 要求類型 有效的統計資訊：平均、最小值 (相容於 P0.0)、最大值 (相同於 p100)、任何介於 p0.0 與 p100 的百分位數

指標	描述
ThrottleCount	對 提出 MediaStore 的HTTP請求數量已調節。 單位：計數 有效維度： • 容器名稱 • 物件群組名稱 • 要求類型 有效的統計資訊：總和

標記 AWS Elemental MediaStore 資源

標籤是一種自訂屬性標籤，可讓您指派或由 AWS 指派給 AWS 資源。每個標籤有兩個部分：

- 標籤鍵 (例如，CostCenter、Environment 或 Project)。標籤鍵會區分大小寫。
- 選用欄位，稱為標籤值 (例如 111122223333 或 Production)。忽略標籤值基本上等同於使用空字串。與標籤鍵相同，標籤值會區分大小寫。

標籤可協助您執行以下操作：

- 識別和組織您的 AWS 資源。許多 AWS 服務支援標籤，因此您可以對來自不同服務的資源指派相同的標籤，以指出資源是相關的。例如，您可以將相同的標籤指派給指派給AWS Elemental MediaLive輸入的 AWS Elemental MediaStore ##。
- 追蹤您的 AWS 成本。您會在 AWS Billing and Cost Management 儀表板上啟用這些標籤。AWS 會使用標籤來分類您的成本，並提供每月成本分配報告給您。如需詳細資訊，請參閱《AWS Billing 使用者指南》中的使用成本分配標籤。

以下各節提供有關 AWS Elemental 標籤的詳細資訊 MediaStore。

AWS Elemental 支援的資源 MediaStore

AWS Elemental MediaStore 支援標記中的下列資源：

- ##

如需新增和管理標籤的詳細資訊，請參閱管理標籤。

AWS Elemental MediaStore 不支援 AWS Identity and Access Management (IAM) 的以標籤為基礎的存取控制功能。

標籤命名和使用慣例

下列基本命名和使用慣例適用於搭配 AWS Elemental MediaStore 資源使用標籤：

- 每個資源的上限為 50 個標籤。
- 對於每一個資源，每個標籤金鑰必須是唯一的，且每個標籤金鑰只能有一個值。
- 最大標籤索引鍵長度為 128 個 UTF-8 形式的 Unicode 字元。
- 最大標籤值長度為 256 個 UTF-8 形式的 Unicode 字元。
- 允許的字元包括可用 UTF-8 表示的英文字母、數字、空格，還有以下特殊字元：. : + = @ _ / - (連字號)。Amazon EC2 資源允許任何字元。
- 標籤鍵與值皆區分大小寫。做為最佳實務，請決定大寫標籤的策略，並一致地在所有資源類型中實作該策略。例如，決定要使用 `Costcenter`、`costcenter` 還是 `CostCenter`，並針對所有標籤使用相同的慣例。避免針對相似的標籤使用不一致的大小寫處理。
- 標籤禁止使用 `aws:` 字首，它保留給 AWS 使用。您不可編輯或刪除具此字首的標籤金鑰或值。具此字首的標籤，不會算在每個資源配額的標籤計數內。

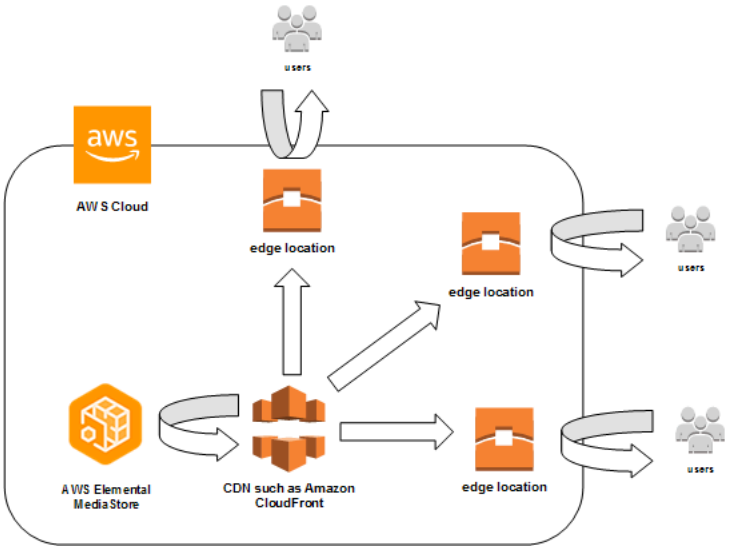
管理標籤

標籤是由資源上的 `Key` 和 `Value` 屬性組成。您可以使用AWS CLI或 MediaStore API 來新增、編輯或刪除這些屬性的值。如需使用標籤的相關資訊，請參閱 AWS Elemental MediaStore API 參考中的以下各節：

- CreateContainer
- ListTagsForResource
- 資源
- TagResource
- UntagResource

使用內容交付網路 (CDN)

您可以使用亞馬遜等內容交付網路 (CDN) CloudFront 來提供存放在 AWS Elemental 中的內容 MediaStore。CDN 是快取影片等內容的全球分佈伺服器。當使用者要求提供您的內容時，CDN 將此要求路由至產生最低延遲的節點。如果您的內容已於該節點中快取，CDN 就會立即提供該內容。如果您的內容目前不在該節點，CDN 會從您的來源 (例如您的容 MediaStore 器) 擷取該內容，並將其散佈給使用者。



主題

- 允許亞馬遜 CloudFront 訪問您的 AWS Elemental MediaStore 容器
- AWS Elemental 與 HTTP 快取 MediaStore的互動

允許亞馬遜 CloudFront 訪問您的 AWS Elemental MediaStore 容器

您可以使用亞馬遜 CloudFront 為您存放在 AWS Elemental 容器中的內容提供服務 MediaStore。您可以下列其中一種方法：

- 使用原始存取控制 (OAC)-(建議) 如果您AWS 區域支援的 OAC 功能，請使用此選項 CloudFront。
- 使用共用 Secrets-如果您AWS 區域不支援的 OAC 功能，請使用此選項 CloudFront。

使用原始存取控制 (OAC)

您可以使用 Amazon 的來源存取控制 (OAC) 功能，透過改善的安全性 CloudFront 來保護 AWS Elemental 來 MediaStore 源的安全性。您可以針對 MediaStore來源的 CloudFront 要求啟用AWS

簽章版本 4 (SIGv4)，並設定何時及是否 CloudFront 應簽署要求。您可以透 CloudFront 過主控台、API、SDK 或 CLI 存取的 OAC 功能，而且使用無需額外費用。

如需搭配使用 OAC 功能的詳細資訊 MediaStore，請參閱 Amazon CloudFront 開發人員指南中的限制對 MediaStore 來源的存取。

使用共用 Secrets

如果您AWS 區域不支援 Amazon 的 OAC 功能 CloudFront，您可以將政策附加到 AWS Elemental MediaStore 容器，以授予讀取權限或更高的權限 CloudFront。

> **Note**
>
> 如果您AWS 區域支援 OAC 功能，我們建議您使用此功能。下列程序需要您設定 MediaStore 並使 CloudFront 用共用密碼，以限制對 MediaStore 容器的存取。若要遵循最佳安全性做法，此手動組態需要定期輪替密碼。透過 MediaStore 原始 OAC，您可以指示使用 SIGv4 簽署 CloudFront 要求，並將其轉寄至 MediaStore 簽名比對，無需使用和輪換機密。這樣可以確保在提供媒體內容之前自動驗證請求，從 MediaStore 而使媒體內容的傳遞更加 CloudFront 簡單，更安全。

允許訪 CloudFront 問您的容器（控制台）

1. 請在以下位置開啟 MediaStore 主控台。 https://console.aws.amazon.com/mediastore/
2. 在 Containers (容器) 頁面上，選擇容器名稱。

   容器詳細資訊頁面隨即出現。
3. 在容器政策區段中，附加授予讀取存取權或更高權限給 Amazon 的政策 CloudFront。

   Example

   下列範例原則類似於透過 HTTPS 進行公開讀取存取的範例原則，符合這些需求，因為它允許`GetObject`任何透過 HTTPS 向您網域提交要求的使用者的`DescribeObject`命令。此外，下列範例原則更能保護您的工作流程，因為只有在要求透過 HTTPS 連線發生且包含正確的 Referer 標頭時，才允許 CloudFront 存取 MediaStore 物件。

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

```
      "Sid": "CloudFrontRead",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "mediastore:GetObject",
        "mediastore:DescribeObject"
      ],
      "Resource": "arn:aws:mediastore:<region>:<owner acct
number>:container/<container name>/*",
      "Condition": {
        "StringEquals": {
          "aws:Referer": "<secretValue>"
        },
        "Bool": {
          "aws:SecureTransport": "true"
        }
      }
    }
]}
```

4. 在 Container CORS policy (容器 CORS 政策) 區段中，指派允許適當存取等級的政策。

> **Note**
>
> 只有在您想要讓使用者存取瀏覽器型播放程式時，CORS 政策才為必要。

5. 請記下列詳細資訊：

   - 指定給容器的資料端點。您可以在 Containers (容器) 頁面的 Info (資訊) 區段中找到這項資訊。在中 CloudFront，資料端點稱為原始網域名稱。
   - 存放物件之容器中的資料夾結構。在中 CloudFront，這稱為原點路徑。請注意，這是選擇性設定。如需有關原始路徑的詳細資訊，請參閱 Amazon CloudFront 開發人員指南。

6. 在中 CloudFront，建立設定為提供 AWS Elemental 內容的發佈 MediaStore。您將需要您在前面步驟中收集的資訊。

將原則附加至 MediaStore 容器之後，您必須設定 CloudFront 為僅針對原始要求使用 HTTPS 連線，並新增具有正確密碼值的自訂標頭。

若 CloudFront 要設定透過 HTTPS 連線存取您的容器，其中包含 Revirer 標頭 (主控台) 的密碼值

1. 開啟主 CloudFront 控台。

2. 在「起源」頁面上，選擇您的 MediaStore來源。
3. 選擇 編輯 。
4. 僅針對通訊協定選擇 HTTPS。
5. 在「新增自訂標頭」區段中，選擇「新增標頭」。
6. 對於「名稱」，選擇「參考者」。對於值，請使用<secretValue>您在容器政策中使用的相同字串。
7. 選擇 [儲存] 並讓變更部署。

AWS Elemental 與 HTTP 快取 MediaStore的互動

AWS Elemental 可 MediaStore 存放物件，以便透過亞馬遜等內容交付網路 (CDN) 正確且有效率地快取這些物件 CloudFront。當一般使用者或 CDN 從中擷取物件時 MediaStore，服務會傳回影響物件快取行為的 HTTP 標頭。HTTP 1.1 快取行為的標準可在 RFC2616 第 13 節中取得)。這些標頭包括：

- **`ETag`** (不可自訂) -實體標籤標題是 MediaStore 傳送回應的唯一識別符。符合標準的 CDN 和網頁瀏覽器會使用此標籤做為快取物件的索引鍵。 MediaStore 上載時，會自動`ETag`為每個物件產生一個。您可以檢視物件的詳細資料，以判斷其 ETag 值。
- **`Last-Modified`**(不可自訂) — 此標頭的值表示修改物件的日期和時間。 MediaStore 上載物件時會自動產生此值。
- **`Cache-Control`** (可自訂) — 此標頭的值會控制物件在 CDN 查看它是否已遭修改前，應該快取的時間長度。當您使用 CLI 或 API 將物件上傳到 MediaStore 容器時，您可以將此標頭設定為任何值。完整一組的有效值會在 HTTP/1.1 文件中加以說明。如果您在上傳物件時未設定此值，則擷取物件時不 MediaStore 會傳回此標頭。

  快取控制標頭常見的使用案例是指定快取物件的持續時間。例如，假設您的視訊資訊清單檔案經常遭編碼器覆寫。您可以將 `max-age` 設為 10，表示該物件應該快取只有 10 秒。或者，假設您有永遠不會遭覆寫的儲存視訊區段。您可以將這個物件的 `max-age` 設定為 31536000 來快取約 1 年的長度。

條件式請求

有條件的請求 MediaStore

MediaStore 對條件式要求 (使用要求標頭 (如 RFC7232 中所述)`If-Modified-Since` 和`If-None-Match`無條件要求的回應完全相同。這意味著，當 MediaStore 收到有效的`GetObject`請求時，即使客戶端已經擁有該對象，服務始終返回該對象。

對 CDN 的條件請求

代表提供內容的 CDN MediaStore 可以透過傳回來處理條件式要求`304 Not Modified`，如 RFC7232 第 4.1 節所述。這表示系統已不需要傳輸完整的物件內容，因為請求者已經有符合條件請求的物件。

CDN (以及與 HTTP/1.1 相容的其他快取) 會將這些決策基於原始服務器轉發的 `ETag` 和 `Cache-Control` 標頭。若要控制 CDN 查詢 MediaStore 原始伺服器以取得重複擷取物件的更新頻率，請在將這些物件上傳至時設定這些物件的`Cache-Control`標頭 MediaStore。

將此服務與 搭配使用 AWS SDK

AWS 軟體開發套件 (SDKs) 適用於許多熱門的程式設計語言。每個 都SDK提供 API、程式碼範例和文件，讓開發人員更輕鬆地以他們偏好的語言建置應用程式。

SDK 文件	代碼範例
AWS SDK for C++	AWS SDK for C++ 程式碼範例
AWS CLI	AWS CLI 程式碼範例
適用於 Go 的 AWS SDK	適用於 Go 的 AWS SDK 程式碼範例
AWS SDK for Java	AWS SDK for Java 程式碼範例
AWS SDK for JavaScript	AWS SDK for JavaScript 程式碼範例
適用於 Kotlin 的 AWS SDK	適用於 Kotlin 的 AWS SDK 程式碼範例
AWS SDK for .NET	AWS SDK for .NET 程式碼範例
AWS SDK for PHP	AWS SDK for PHP 程式碼範例
AWS Tools for PowerShell	PowerShell 程式碼範例的工具
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) 程式碼範例
AWS SDK for Ruby	AWS SDK for Ruby 程式碼範例
適用於 Rust 的 AWS SDK	適用於 Rust 的 AWS SDK 程式碼範例
適用於 SAP ABAP 的 AWS SDK	適用於 SAP ABAP 的 AWS SDK 程式碼範例
適用於 Swift 的 AWS SDK	適用於 Swift 的 AWS SDK 程式碼範例

如需此服務的特定範例，請參閱 MediaStore 使用 的程式碼範例 AWS SDKs。

可用性範例

找不到所需的內容嗎？ 請使用本頁面底部的提供意見回饋連結申請程式碼範例。

MediaStore 使用 的程式碼範例 AWS SDKs

下列程式碼範例示範如何 MediaStore 搭配 AWS 軟體開發套件 () 使用 SDK。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然 動作會示範如何呼叫個別服務函數，但您可以在其相關案例的內容中看到動作。

如需開發人員指南和程式碼範例的完整清單 AWS SDK，請參閱 將此服務與 搭配使用 AWS SDK。本主題也包含入門的相關資訊，以及先前SDK版本的詳細資訊。

程式碼範例

- MediaStore 使用 的基本範例 AWS SDKs
  - MediaStore 使用 的動作 AWS SDKs
    - CreateContainer 搭配 AWS SDK或 使用 CLI
    - DeleteContainer 搭配 AWS SDK或 使用 CLI
    - DeleteObject 搭配 使用 AWS SDK
    - DescribeContainer 搭配 AWS SDK或 使用 CLI
    - GetObject 搭配 AWS SDK或 使用 CLI
    - ListContainers 搭配 AWS SDK或 使用 CLI
    - PutObject 搭配 AWS SDK或 使用 CLI

MediaStore 使用 的基本範例 AWS SDKs

下列程式碼範例示範如何 AWS Elemental MediaStore 搭配 AWS 使用 的基本概念SDKs。

範例

- MediaStore 使用 的動作 AWS SDKs
  - CreateContainer 搭配 AWS SDK或 使用 CLI
  - DeleteContainer 搭配 AWS SDK或 使用 CLI
  - DeleteObject 搭配 使用 AWS SDK
  - DescribeContainer 搭配 AWS SDK或 使用 CLI
  - GetObject 搭配 AWS SDK或 使用 CLI
  - ListContainers 搭配 AWS SDK或 使用 CLI
  - PutObject 搭配 AWS SDK或 使用 CLI

MediaStore 使用 的動作 AWS SDKs

下列程式碼範例示範如何使用 執行個別 MediaStore動作 AWS SDKs。每個範例都包含一個 連結 GitHub，您可以在其中找到設定和執行程式碼的指示。

下列範例僅包含最常使用的動作。如需完整清單，請參閱 AWS Elemental MediaStore API 參考。

範例

- CreateContainer 搭配 AWS SDK或 使用 CLI
- DeleteContainer 搭配 AWS SDK或 使用 CLI
- DeleteObject 搭配 使用 AWS SDK
- DescribeContainer 搭配 AWS SDK或 使用 CLI
- GetObject 搭配 AWS SDK或 使用 CLI
- ListContainers 搭配 AWS SDK或 使用 CLI
- PutObject 搭配 AWS SDK或 使用 CLI

`CreateContainer` 搭配 AWS SDK或 使用 CLI

下列程式碼範例示範如何使用 `CreateContainer`。

CLI

AWS CLI

建立容器

下列`create-container`範例會建立新的空容器。

```
aws mediastore create-container --container-name ExampleContainer
```

輸出：

```
{
    "Container": {
        "AccessLoggingEnabled": false,
        "CreationTime": 1563557265,
        "Name": "ExampleContainer",
        "Status": "CREATING",
```

```
        "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer"
    }
}
```

如需詳細資訊，請參閱 AWS 元素 MediaStore 使用者指南中的建立容器。

- 如需API詳細資訊，請參閱 AWS CLI 命令參考CreateContainer中的 。

Java

SDK 適用於 Java 2.x

> **Note**
>
> 還有更多功能 GitHub。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
import software.amazon.awssdk.services.mediastore.MediaStoreClient;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.mediastore.model.CreateContainerRequest;
import software.amazon.awssdk.services.mediastore.model.CreateContainerResponse;
import software.amazon.awssdk.services.mediastore.model.MediaStoreException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateContainer {
    public static long sleepTime = 10;

    public static void main(String[] args) {
        final String usage = """

                Usage:    <containerName>
```

```
                Where:
                   containerName - The name of the container to create.
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String containerName = args[0];
        Region region = Region.US_EAST_1;
        MediaStoreClient mediaStoreClient = MediaStoreClient.builder()
                .region(region)
                .build();

        createMediaContainer(mediaStoreClient, containerName);
        mediaStoreClient.close();
    }

    public static void createMediaContainer(MediaStoreClient mediaStoreClient,
String containerName) {
        try {
            CreateContainerRequest containerRequest =
CreateContainerRequest.builder()
                    .containerName(containerName)
                    .build();

            CreateContainerResponse containerResponse =
mediaStoreClient.createContainer(containerRequest);
            String status = containerResponse.container().status().toString();
            while (!status.equalsIgnoreCase("Active")) {
                status = DescribeContainer.checkContainer(mediaStoreClient,
containerName);
                System.out.println("Status - " + status);
                Thread.sleep(sleepTime * 1000);
            }

            System.out.println("The container ARN value is " +
containerResponse.container().arn());
            System.out.println("Finished ");

        } catch (MediaStoreException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
```

```
        }
    }
}
```

- 如需API詳細資訊，請參閱 AWS SDK for Java 2.x API 參考CreateContainer中的 。

如需開發人員指南和程式碼範例的完整清單 AWS SDK，請參閱 將此服務與 搭配使用 AWS SDK。本主題也包含入門的相關資訊，以及先前SDK版本的詳細資訊。

`DeleteContainer` 搭配 AWS SDK或 使用 CLI

下列程式碼範例示範如何使用 `DeleteContainer`。

CLI

AWS CLI

刪除容器

下列`delete-container`範例會刪除指定的容器。只有當容器沒有任何物件時，您才可以將該容器刪除。

```
aws mediastore delete-container \
    --container-name=ExampleLiveDemo
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS 元素 MediaStore 使用者指南中的刪除容器。

- 如需API詳細資訊，請參閱 AWS CLI 命令參考DeleteContainer中的 。

Java

SDK 適用於 Java 2.x

> **Note**
>
> 還有更多功能 GitHub。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
import software.amazon.awssdk.services.mediastore.MediaStoreClient;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.mediastore.model.CreateContainerRequest;
import software.amazon.awssdk.services.mediastore.model.CreateContainerResponse;
import software.amazon.awssdk.services.mediastore.model.MediaStoreException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateContainer {
    public static long sleepTime = 10;

    public static void main(String[] args) {
        final String usage = """

                Usage:    <containerName>

                Where:
                   containerName - The name of the container to create.
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String containerName = args[0];
        Region region = Region.US_EAST_1;
        MediaStoreClient mediaStoreClient = MediaStoreClient.builder()
                .region(region)
                .build();

        createMediaContainer(mediaStoreClient, containerName);
        mediaStoreClient.close();
    }
```

```
    public static void createMediaContainer(MediaStoreClient mediaStoreClient,
String containerName) {
        try {
            CreateContainerRequest containerRequest =
CreateContainerRequest.builder()
                    .containerName(containerName)
                    .build();

            CreateContainerResponse containerResponse =
mediaStoreClient.createContainer(containerRequest);
            String status = containerResponse.container().status().toString();
            while (!status.equalsIgnoreCase("Active")) {
                status = DescribeContainer.checkContainer(mediaStoreClient,
containerName);
                System.out.println("Status - " + status);
                Thread.sleep(sleepTime * 1000);
            }

            System.out.println("The container ARN value is " +
containerResponse.container().arn());
            System.out.println("Finished ");

        } catch (MediaStoreException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

- 如需API詳細資訊，請參閱 AWS SDK for Java 2.x API 參考DeleteContainer中的 。

如需開發人員指南和程式碼範例的完整清單 AWS SDK，請參閱 將此服務與 搭配使用 AWS SDK。本主題也包含入門的相關資訊，以及先前SDK版本的詳細資訊。

`DeleteObject` 搭配 使用 AWS SDK

下列程式碼範例示範如何使用 `DeleteObject`。

Java

SDK 適用於 Java 2.x

> **Note**
>
> 還有更多功能 GitHub。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.mediastore.MediaStoreClient;
import software.amazon.awssdk.services.mediastore.model.DescribeContainerRequest;
import
 software.amazon.awssdk.services.mediastore.model.DescribeContainerResponse;
import software.amazon.awssdk.services.mediastoredata.MediaStoreDataClient;
import software.amazon.awssdk.services.mediastoredata.model.DeleteObjectRequest;
import
 software.amazon.awssdk.services.mediastoredata.model.MediaStoreDataException;
import java.net.URI;
import java.net.URISyntaxException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteObject {
    public static void main(String[] args) throws URISyntaxException {
        final String usage = """

                Usage:    <completePath> <containerName>

                Where:
                   completePath - The path (including the container) of the item
 to delete.
                   containerName - The name of the container.
                """;
```

```
        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String completePath = args[0];
        String containerName = args[1];
        Region region = Region.US_EAST_1;
        URI uri = new URI(getEndpoint(containerName));

        MediaStoreDataClient mediaStoreData = MediaStoreDataClient.builder()
                .endpointOverride(uri)
                .region(region)
                .build();

        deleteMediaObject(mediaStoreData, completePath);
        mediaStoreData.close();
    }

    public static void deleteMediaObject(MediaStoreDataClient mediaStoreData,
String completePath) {
        try {
            DeleteObjectRequest deleteObjectRequest =
DeleteObjectRequest.builder()
                    .path(completePath)
                    .build();

            mediaStoreData.deleteObject(deleteObjectRequest);

        } catch (MediaStoreDataException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    private static String getEndpoint(String containerName) {
        Region region = Region.US_EAST_1;
        MediaStoreClient mediaStoreClient = MediaStoreClient.builder()
                .region(region)
                .build();

        DescribeContainerRequest containerRequest =
DescribeContainerRequest.builder()
```

```
                .containerName(containerName)
                .build();

        DescribeContainerResponse response =
 mediaStoreClient.describeContainer(containerRequest);
        mediaStoreClient.close();
        return response.container().endpoint();
    }
}
```

- 如需API詳細資訊，請參閱 AWS SDK for Java 2.x API 參考DeleteObject中的 。

如需開發人員指南和程式碼範例的完整清單 AWS SDK，請參閱 將此服務與 搭配使用 AWS SDK。本主題也包含入門的相關資訊，以及先前SDK版本的詳細資訊。

`DescribeContainer` 搭配 AWS SDK或 使用 CLI

下列程式碼範例示範如何使用 `DescribeContainer`。

CLI

AWS CLI

檢視容器的詳細資訊

下列`describe-container`範例顯示指定容器的詳細資訊。

```
aws mediastore describe-container \
    --container-name ExampleContainer
```

輸出：

```
{
    "Container": {
        "CreationTime": 1563558086,
        "AccessLoggingEnabled": false,
        "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer",
        "Status": "ACTIVE",
        "Name": "ExampleContainer",
```

```
        "Endpoint": "https://aaabbbcccdddee.data.mediastore.us-
west-2.amazonaws.com"
    }
}
```

如需詳細資訊，請參閱 AWS 元素 MediaStore 使用者指南中的檢視容器的詳細資訊。

- 如需API詳細資訊，請參閱 AWS CLI 命令參考DescribeContainer中的 。

Java

SDK 適用於 Java 2.x

> **Note**
>
> 還有更多功能 GitHub。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.mediastore.MediaStoreClient;
import software.amazon.awssdk.services.mediastore.model.DescribeContainerRequest;
import
 software.amazon.awssdk.services.mediastore.model.DescribeContainerResponse;
import software.amazon.awssdk.services.mediastore.model.MediaStoreException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeContainer {

    public static void main(String[] args) {
        final String usage = """

                Usage:    <containerName>
```

```
                Where:
                   containerName - The name of the container to describe.
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String containerName = args[0];
        Region region = Region.US_EAST_1;
        MediaStoreClient mediaStoreClient = MediaStoreClient.builder()
                .region(region)
                .build();

        System.out.println("Status is " + checkContainer(mediaStoreClient,
 containerName));
        mediaStoreClient.close();
    }

    public static String checkContainer(MediaStoreClient mediaStoreClient, String
 containerName) {
        try {
            DescribeContainerRequest describeContainerRequest =
 DescribeContainerRequest.builder()
                    .containerName(containerName)
                    .build();

            DescribeContainerResponse containerResponse =
 mediaStoreClient.describeContainer(describeContainerRequest);
            System.out.println("The container name is " +
 containerResponse.container().name());
            System.out.println("The container ARN is " +
 containerResponse.container().arn());
            return containerResponse.container().status().toString();

        } catch (MediaStoreException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- 如需API詳細資訊，請參閱 AWS SDK for Java 2.x API 參考DescribeContainer中的 。

如需開發人員指南和程式碼範例的完整清單 AWS SDK，請參閱 將此服務與 搭配使用 AWS SDK。本主題也包含入門的相關資訊，以及先前SDK版本的詳細資訊。

`GetObject` 搭配 AWS SDK或 使用 CLI

下列程式碼範例示範如何使用 `GetObject`。

CLI

AWS CLI

下載物件

下列get-object範例會將物件下載至指定的端點。

```
aws mediastore-data get-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path=/folder_name/README.md README.md
```

輸出：

```
{
    "ContentLength": "2307346",
    "ContentType": "image/jpeg",
    "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
    "StatusCode": 200
}
```

下載部分物件

下列get-object範例會將物件的一部分下載至指定的端點。

```
aws mediastore-data get-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path /folder_name/README.md \
    --range="bytes=0-100" README2.md
```

輸出：

```
{
    "StatusCode": 206,
    "ContentRange": "bytes 0-100/2307346",
    "ContentLength": "101",
    "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
    "ContentType": "image/jpeg",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
}
```

如需詳細資訊，請參閱 AWS 元素 MediaStore 使用者指南中的下載物件。

- 如需API詳細資訊，請參閱 AWS CLI 命令參考GetObject中的 。

Java

SDK 適用於 Java 2.x

> **Note**
>
> 還有更多功能 GitHub。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
import software.amazon.awssdk.core.ResponseInputStream;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.mediastore.MediaStoreClient;
import software.amazon.awssdk.services.mediastore.model.DescribeContainerRequest;
import
 software.amazon.awssdk.services.mediastore.model.DescribeContainerResponse;
import software.amazon.awssdk.services.mediastoredata.MediaStoreDataClient;
import software.amazon.awssdk.services.mediastoredata.model.GetObjectRequest;
import software.amazon.awssdk.services.mediastoredata.model.GetObjectResponse;
import
 software.amazon.awssdk.services.mediastoredata.model.MediaStoreDataException;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.net.URI;
import java.net.URISyntaxException;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class GetObject {
    public static void main(String[] args) throws URISyntaxException {
        final String usage = """

                Usage:    <completePath> <containerName> <savePath>

                Where:
                   completePath - The path of the object in the container (for
 example, Videos5/sampleVideo.mp4).
                   containerName - The name of the container.
                   savePath - The path on the local drive where the file is
 saved, including the file name (for example, C:/AWS/myvid.mp4).
                """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String completePath = args[0];
        String containerName = args[1];
        String savePath = args[2];

        Region region = Region.US_EAST_1;
        URI uri = new URI(getEndpoint(containerName));
        MediaStoreDataClient mediaStoreData = MediaStoreDataClient.builder()
                .endpointOverride(uri)
                .region(region)
                .build();

        getMediaObject(mediaStoreData, completePath, savePath);
        mediaStoreData.close();
    }
```

```
    public static void getMediaObject(MediaStoreDataClient mediaStoreData, String
 completePath, String savePath) {

        try {
            GetObjectRequest objectRequest = GetObjectRequest.builder()
                    .path(completePath)
                    .build();

            // Write out the data to a file.
            ResponseInputStream<GetObjectResponse> data =
 mediaStoreData.getObject(objectRequest);
            byte[] buffer = new byte[data.available()];
            data.read(buffer);

            File targetFile = new File(savePath);
            OutputStream outStream = new FileOutputStream(targetFile);
            outStream.write(buffer);
            System.out.println("The data was written to " + savePath);

        } catch (MediaStoreDataException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    private static String getEndpoint(String containerName) {
        Region region = Region.US_EAST_1;
        MediaStoreClient mediaStoreClient = MediaStoreClient.builder()
                .region(region)
                .build();

        DescribeContainerRequest containerRequest =
 DescribeContainerRequest.builder()
                .containerName(containerName)
                .build();

        DescribeContainerResponse response =
 mediaStoreClient.describeContainer(containerRequest);
        return response.container().endpoint();
    }
}
```

- 如需API詳細資訊，請參閱 AWS SDK for Java 2.x API 參考GetObject中的 。

如需開發人員指南和程式碼範例的完整清單 AWS SDK，請參閱 將此服務與 搭配使用 AWS SDK。本主題也包含入門的相關資訊，以及先前SDK版本的詳細資訊。

`ListContainers` 搭配 AWS SDK或 使用 CLI

下列程式碼範例示範如何使用 `ListContainers`。

CLI

AWS CLI

檢視容器清單

下列`list-containers`範例顯示與您 帳戶相關聯的所有容器清單。

```
aws mediastore list-containers
```

輸出：

```
{
    "Containers": [
        {
            "CreationTime": 1505317931,
            "Endpoint": "https://aaabbbcccdddee.data.mediastore.us-
west-2.amazonaws.com",
            "Status": "ACTIVE",
            "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleLiveDemo",
            "AccessLoggingEnabled": false,
            "Name": "ExampleLiveDemo"
        },
        {
            "CreationTime": 1506528818,
            "Endpoint": "https://fffggghhhiiijj.data.mediastore.us-
west-2.amazonaws.com",
            "Status": "ACTIVE",
            "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer",
            "AccessLoggingEnabled": false,
            "Name": "ExampleContainer"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS 元素 MediaStore 使用者指南中的檢視容器清單。

- 如需API詳細資訊，請參閱 AWS CLI 命令參考ListContainers中的 。

Java

SDK 適用於 Java 2.x

> **Note**
>
> 還有更多功能 GitHub。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.mediastore.MediaStoreClient;
import software.amazon.awssdk.services.mediastore.model.Container;
import software.amazon.awssdk.services.mediastore.model.ListContainersResponse;
import software.amazon.awssdk.services.mediastore.model.MediaStoreException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListContainers {

    public static void main(String[] args) {

        Region region = Region.US_EAST_1;
        MediaStoreClient mediaStoreClient = MediaStoreClient.builder()
                .region(region)
                .build();

        listAllContainers(mediaStoreClient);
        mediaStoreClient.close();
```

```
    }

    public static void listAllContainers(MediaStoreClient mediaStoreClient) {
        try {
            ListContainersResponse containersResponse =
 mediaStoreClient.listContainers();
            List<Container> containers = containersResponse.containers();
            for (Container container : containers) {
                System.out.println("Container name is " + container.name());
            }

        } catch (MediaStoreException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- 如需API詳細資訊，請參閱 AWS SDK for Java 2.x API 參考ListContainers中的 。

如需開發人員指南和程式碼範例的完整清單 AWS SDK，請參閱 將此服務與 搭配使用 AWS SDK。本主題也包含入門的相關資訊，以及先前SDK版本的詳細資訊。

`PutObject` 搭配 AWS SDK或 使用 CLI

下列程式碼範例示範如何使用 `PutObject`。

CLI

AWS CLI

上傳物件

下列`put-object`範例會將物件上傳至指定的容器。您可以指定資料夾路徑，其中物件將儲存在容器內。如果資料夾已存在， AWS Elemental 會將物件 MediaStore 存放在資料夾中。如果資料夾不存在，服務會建立資料夾，然後將物件存放在資料夾中。

```
aws mediastore-data put-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --body README.md \
    --path /folder_name/README.md \
    --cache-control "max-age=6, public" \
```

```
    --content-type binary/octet-stream
```

輸出：

```
{
    "ContentSHA256":
 "74b5fdb517f423ed750ef214c44adfe2be36e37d861eafe9c842cbe1bf387a9d",
    "StorageClass": "TEMPORAL",
    "ETag": "af3e4731af032167a106015d1f2fe934e68b32ed1aa297a9e325f5c64979277b"
}
```

如需詳細資訊，請參閱 AWS 元素 MediaStore 使用者指南中的上傳物件。

- 如需API詳細資訊，請參閱 AWS CLI 命令參考PutObject中的 。

Java

SDK for Java 2.x

> **Note**
>
> 還有更多功能 GitHub。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.mediastore.MediaStoreClient;
import software.amazon.awssdk.services.mediastoredata.MediaStoreDataClient;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.services.mediastoredata.model.PutObjectRequest;
import
 software.amazon.awssdk.services.mediastoredata.model.MediaStoreDataException;
import software.amazon.awssdk.services.mediastoredata.model.PutObjectResponse;
import software.amazon.awssdk.services.mediastore.model.DescribeContainerRequest;
import
 software.amazon.awssdk.services.mediastore.model.DescribeContainerResponse;
import java.io.File;
import java.net.URI;
import java.net.URISyntaxException;

/**
 * Before running this Java V2 code example, set up your development
```

```
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class PutObject {
    public static void main(String[] args) throws URISyntaxException {
        final String USAGE = """

                To run this example, supply the name of a container, a file
 location to use, and path in the container\s

                Ex: <containerName> <filePath> <completePath>
                """;

        if (args.length < 3) {
            System.out.println(USAGE);
            System.exit(1);
        }

        String containerName = args[0];
        String filePath = args[1];
        String completePath = args[2];

        Region region = Region.US_EAST_1;
        URI uri = new URI(getEndpoint(containerName));
        MediaStoreDataClient mediaStoreData = MediaStoreDataClient.builder()
                .endpointOverride(uri)
                .region(region)
                .build();

        putMediaObject(mediaStoreData, filePath, completePath);
        mediaStoreData.close();
    }

    public static void putMediaObject(MediaStoreDataClient mediaStoreData, String
 filePath, String completePath) {
        try {
            File myFile = new File(filePath);
            RequestBody requestBody = RequestBody.fromFile(myFile);

            PutObjectRequest objectRequest = PutObjectRequest.builder()
```

```
                    .path(completePath)
                    .contentType("video/mp4")
                    .build();

            PutObjectResponse response = mediaStoreData.putObject(objectRequest,
 requestBody);
            System.out.println("The saved object is " +
 response.storageClass().toString());

        } catch (MediaStoreDataException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static String getEndpoint(String containerName) {

        Region region = Region.US_EAST_1;
        MediaStoreClient mediaStoreClient = MediaStoreClient.builder()
                .region(region)
                .build();

        DescribeContainerRequest containerRequest =
 DescribeContainerRequest.builder()
                .containerName(containerName)
                .build();

        DescribeContainerResponse response =
 mediaStoreClient.describeContainer(containerRequest);
        return response.container().endpoint();
    }
}
```

- 如需API詳細資訊，請參閱 AWS SDK for Java 2.x API 參考PutObject中的 。

如需開發人員指南和程式碼範例的完整清單 AWS SDK，請參閱 將此服務與 搭配使用 AWS SDK。本主題也包含入門的相關資訊，以及先前SDK版本的詳細資訊。

AWS Elemental 中的配額 MediaStore

Service Quotas 主控台提供有關 AWS Elemental vice MediaStore 配額的相關資訊。除了檢視預設配額之外，您還可以使用 Service Quotas 主控台來請求增加配額以取得可調整配額。

下表說明 AWS Elemental 中的配額 (先前稱為限制) MediaStore。配額是您 AWS 帳戶的服務資源或操作數目最大值。

> **Note**
>
> 若要將配額指派給帳戶內的個別容器，請聯絡 AWS Support 或您的客戶經理。此選項可協助您在容器之間劃分帳戶層級限制，以防止一個容器使用您的整個配額。

資源或操作	預設配額	說明
容器	100	此帳戶中可建立的容器數量上限。
資料夾層級	10	容器中可建立的資料夾層級數量上限。您可以視需要建立任何數量的資料夾，只要容器中的巢狀層級不超過 10 層。
資料夾	無限制	您可以視需要建立任何數量的資料夾，只要容器中的巢狀層級不超過 10 層。
物件大小	25 MB	單一物件的檔案大小上限。
物件	無限制	您可以將任意數量的物件上傳到帳戶中的資料夾或容器。
DeleteObject API 請求的速率	100	您每秒可以提出的操作請求上限數量。其他請求會受到調節。 您可以要求增加配額。
DescribeObject API 請求的速率	1,000	您每秒可以提出的操作請求上限數量。其他請求會受到調節。 您可以要求增加配額。

資源或操作	預設配額	說明
標準上傳可用性的 GetObjectAPI 要求比率	1,000	您每秒可以提出的操作請求上限數量。其他請求會受到調節。 您可以要求增加配額。
串流上傳可用性的 GetObjectAPI 要求比率	25	您每秒可以提出的操作請求上限數量。其他請求會受到調節。 您可以要求增加配額。
ListItems API 請求的速率	5	您每秒可以提出的操作請求上限數量。其他請求會受到調節。 您可以要求增加配額。
區塊傳輸編碼的 PutObjectAPI 要求率 (也稱為串流上傳可用性)	10	您每秒可以提出的操作請求上限數量。其他請求會受到調節。 您可以要求增加配額。在請求中，指定請求的 TPS 和平均物件大小。
標準上傳可用性的 PutObjectAPI 要求比率	100	您每秒可以提出的操作請求上限數量。其他請求會受到調節。 您可以要求增加配額。在請求中，指定請求的 TPS 和平均物件大小。
指標政策中的規則	10	您可以包含在指標政策中的規則數量上限。
物件生命週期政策中的規則	10	您可以包含在物件生命週期政策中包含的規則數量上限。

AWS Elemental MediaStore 相關資訊

下表列出您在使用 AWS Elemental 時發現有用的相關資源 MediaStore。

- 課程和研討會 — 連結至以角色為基礎的專門課程以及自主進度實驗室，協助加強您的AWS技能，並取得實際體驗。
- AWS開發人員中心 — 研究教學課程、下載工具，以及了解AWS開發人員活動。
- AWS開發人員工具 – 連結至開發人員工具、軟體開發人員工具、開發人員工具、軟體開發人員工具、軟體開發人員AWS工具、
- 入門資源中心 — 了解如何設定AWS 帳戶、加入AWS社群，以及啟動您的第一個應用程式。
- 實用的教學課 step-by-step 程-按照啟動您的第一個應用程式AWS。
- AWS白皮書 — 連結至完整的技術AWS白皮書清單，其中涵蓋了架構、安全和成本等主題，並由AWS解決方案架構師或其他技術專家撰寫。
- AWS Support 中心 – 建立和管理您的 AWS Support 案例的中心。這也包含與其他實用資源的連結，例如論壇、技術常見問答集、服務運作狀態以及 AWS Trusted Advisor。
- Support— 有關的資訊的主要網頁Support，它是快速回應支援頻道 one-on-one，可協助您在雲端中建置並執行應用程式。
- 聯絡我們 – 查詢有關 AWS 帳單、帳戶、事件、濫用與其他問題的聯絡中心。
- AWS 網站條款 – 我們的著作權與商標；您的帳戶、授權與網站存取；以及其他主題的詳細資訊。

使用者指南的文件歷史記錄

下表說明此版本 AWS Elemental 的文件 MediaStore。如需有關本文件更新的通知，您可以訂閱RSS摘要。

變更	描述	日期
支援通知結束	支援終止通知：在 2025 年 11 月 13 日， AWS 將停止對 AWS Elemental 的支援 MediaStore。2025 年 11 月 13 日後，您將無法再存取 MediaStore 主控台或 MediaStore 資源。如需詳細資訊，請造訪此部落格文章 。	2024 年 11 月 12 日
原始伺服器存取控制 （OAC） 改進	新增了如何OAC搭配 AWS Elemental 使用 的相關資訊 MediaStore。	2023 年 4 月 17 日
配額更新	更正 的配額值和描述`Rules in a Metric Policy`。	2022 年 10 月 25 日
ExpiresAt 欄位	存取日誌現在包含一個`ExpiresAt` 欄位，根據容器生命週期政策中的暫時性資料規則來指示物件的過期日期和時間。	2020 年 7 月 16 日
生命週期轉換規則	您現在可以將生命週期轉換規則新增到物件的生命週期政策中，將物件設為在到達一定存留期後移動到不常存取 (IA) 儲存體方案。	2020 年 4 月 20 日
空容器	您現在可以一次刪除容器內的所有物件。	2020 年 4 月 7 日

支援 Amazon CloudWatch 指標	您可以設定指標政策，以指定哪些指標 MediaStore 會傳送至 CloudWatch。	2020 年 3 月 30 日
刪除物件規則中的萬用字元	在物件生命週期政策中，您現在可以在刪除物件規則中使用萬用字元。這可讓您根據要在特定天數後刪除服務的檔案名稱或副檔名來指定檔案。	2019 年 12 月 20 日
物件生命週期政策	您現在可以將規則新增至物件生命週期政策，以秒為單位指定過期時間。	2019 年 9 月 13 日
AWS CloudFormation 支援	您現在可以使用 AWS CloudFormation 範本自動建立容器。 AWS CloudFormation 範本會管理五個API動作的資料：建立容器、設定存取記錄、更新預設容器政策、新增跨來源資源共用（CORS）政策，以及新增物件生命週期政策。	2019 年 5 月 17 日
串流上傳可用性的配額	對於具有串流上傳可用性的物件（物件區塊傳輸），`PutObject` 操作不得超過 10TPS，`GetObject` 操作不得超過 25TPS。	2019 年 4 月 8 日
物件的區塊傳輸	新增對物件區塊傳輸的支援。此功能可讓您指定物件可在完全上傳前供下載使用。	2019 年 4 月 5 日
存取日誌	AWS Elemental MediaStore 現在支援存取記錄，提供對容器中物件提出之請求的詳細記錄。	2019 年 2 月 25 日

物件生命週期政策	新增對物件生命週期政策的支援，此政策可管理目前容器中物件的過期日期。	2018 年 12 月 12 日
物件大小配額增加	物件大小的配額現在是 25 MB。	2018 年 10 月 10 日
物件大小配額增加	物件大小的配額現在是 20 MB。	2018 年 9 月 6 日
AWS CloudTrail 整合	CloudTrail 整合內容已更新，以符合 CloudTrail 服務的最新變更。	2018 年 7 月 12 日
CDN 協同合作	新增了有關如何將 AWS Elemental MediaStore 與內容交付網路（CDN）搭配使用的相關資訊，例如 Amazon CloudFront。	2018 年 4 月 14 日
CORS 組態	AWS Elemental MediaStore 現在支援跨來源資源共用（CORS），允許載入某個網域的用戶端 Web 應用程式與不同網域的資源互動。	2018 年 2 月 7 日
新的服務與指南	這是影片原始伺服器和儲存服務的初始版本、AWSElemental MediaStore和 AWS Elemental MediaStore 使用者指南 。	2017 年 11 月 27 日

Note

- AWS 媒體服務並非設計用於與應用程式或需要不安全效能的情況下使用，例如生命安全操作、導航或通訊系統、空中流量控制或生命支援機器，其中服務的無法使用、中斷或故障可能會導致死亡、人員受傷、財產損壞或環境損壞。

AWS 詞彙表

如需最新的 AWS 術語，請參閱《AWS 詞彙表 參考》中的 AWS 詞彙表。