aws

Guía del usuario

AWS Elemental MediaStore

AWS Elemental MediaStore: Guía del usuario

Table of Contents

Aviso de fin de soporte: el 13 de noviembre de 2025, AWS suspenderemos el soporte para AWS Elemental MediaStore. Después del 13 de noviembre de 2025, ya no podrá acceder a la MediaStore consola ni a MediaStore los recursos. Para obtener más información, visite esta publicación del blog.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.

¿Qué es AWS Elemental Elemental Elemental MediaStore Elem

AWS Elemental Elemental MediaStore , un servicio de distribución y almacenamiento de vídeo que ofrece el alto desempeño y la coherencia inmediata necesarios para distribuir contenido en directo. Con MediaStore, es posible administrar recursos de vídeo como objetos en contenedores para crear flujos de trabajo de recursos multimedia fiables y basados en la nube.

Para utilizar el servicio, se cargan objetos desde un origen, como, por ejemplo, un codificador o una fuente de datos, a un contenedor que se crea en MediaStore.

MediaStore es una opción ideal para almacenar archivos de vídeo fragmentados cuando se necesita un gran nivel de coherencia, baja latencia de lectura y escritura, y la capacidad para gestionar un gran número de solicitudes simultáneas. Si no difunde vídeos en directo por streaming, contemple la posibilidad de utilizar Amazon Simple Storage Service (Amazon S3) como alternativa.

Temas

MediaStore Conceptos y terminología de AWS Elemental Elemental Elemental

ARN

Un Nombre de recurso de Amazon.

Cuerpo

Los datos que se van a cargar en un objeto.

Rango (de bytes)

Un subconjunto de datos de un objeto a los que se tiene acceso. Para obtener más información, consulte intervalo de la especificación HTTP.

Contenedor

Un espacio de nombres que contiene objetos. Un contenedor tiene un punto de enlace que se puede utilizar para escribir y recuperar objetos y asociar políticas de acceso.

Punto de conexión

Un punto de entrada al MediaStore servicio, que se especifica por medio de una URL raíz HTTPS.

ETag

Una etiqueta de entidad, que es un hash de los datos del objeto.

Carpeta

Una división de un contenedor. Una carpeta puede contener objetos y otras carpetas.

Elemento

Un término que se usa para hacer referencia a los objetos y las carpetas.

Objeto

Un recurso, similar a un objeto de Amazon S3. Los objetos son las entidades fundamentales que se almacenan en MediaStore. El servicio acepta todos los tipos de archivos.

Servicio de distribución

MediaStore se considera un servicio de distribución porque es el punto de distribución para la entrega de contenido multimedia.

Ruta

Un identificador único para un objeto o una carpeta, que indica su ubicación en el contenedor.

Parte

Un subconjunto de datos (fragmento) de un objeto.

Política

Una política de IAM.

Recurso

Una entidad en AWS con la que puede trabajar. A cada recurso de AWS se le asigna un nombre de recurso de Amazon (ARN) que actúa como un identificador único. En MediaStore, este es el recurso y su formato de ARN:

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

Servicios relacionados

- Amazon CloudFront es un servicio de red de entrega de contenido (CDN) global que entrega datos y videos de forma segura a los espectadores. Utilice CloudFront para enviar contenido con el mejor desempeño posible. Para obtener más información, consulte la Guía para CloudFront desarrolladores de Amazon.
- AWS CloudFormationes un servicio que ayuda a modelar y configurar AWS los recursos de. Puede crear una plantilla que describa todos los AWS recursos de que desea (como MediaStore contenedores) y AWS CloudFormation se encargará del aprovisionamiento y la configuración de dichos recursos. No es necesario crear y configurar individualmente AWS los recursos de ni averiguar qué depende de qué; se AWS CloudFormation encarga de todo eso. Para obtener más información, consulte la Guía del usuario de AWS CloudFormation.
- AWS CloudTrailes un servicio que le permite monitorizar las llamadas realizadas a la CloudTrail API para su cuenta, incluidas las llamadas realizadas mediante la consola de administración AWS CLI, la y otros servicios de. Para obtener más información, consulte la Guía del usuario de AWS CloudTrail.
- Amazon CloudWatch es un servicio de supervisión de los recursos de la AWS nube y las aplicaciones en las que se ejecuta AWS. Utilice CloudWatch Eventos para realizar un seguimiento de los cambios en el estado de los contenedores y los objetos en MediaStore. Para obtener más información, consulte la CloudWatch documentación de Amazon.
- AWS Identity and Access Management (IAM) es un servicio web que ayuda a controlar de forma segura el acceso de los usuarios a AWS los recursos de. Utilice IAM para controlar quién puede usar AWS los recursos de (autenticación) y cuáles de ellos pueden usar los usuarios y cómo pueden hacerlo (autorización). Para obtener más información, consulte Configuración de AWS Elemental MediaStore.
- Amazon Simple Storage Service (Amazon S3) es un almacenamiento de objetos creado para almacenar y recuperar cualquier cantidad de datos desde cualquier lugar. Para obtener más información, consulte la documentación de Amazon S3.

Acceso a AWS Elemental Elemental Elem MediaStore

Puede obtener acceso MediaStore mediante cualquiera de los siguientes métodos:

- Consola de administración de AWS: los procedimientos de esta guía explican cómo utilizar la consola de administración de AWS para realizar tareas para MediaStore. Para acceder a MediaStore través de la consola:

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

- AWS Command Line Interface Para obtener más información, consulte la Guía de usuario de AWS Command Line Interface. Para acceder MediaStore mediante el punto de conexión CLI:

  ```
  aws mediastore
  ```

- MediaStore API de: si utiliza un lenguaje de programación para el que no exista un SDK, consulte la Referencia de la AWS Elemental MediaStore API para obtener información acerca de las acciones de API y cómo realizar solicitudes de API. Para acceder MediaStore mediante el punto de conexión de la API de REST:

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

- AWS SDKs: si utiliza un lenguaje de programación para el que AWS proporciona un SDK, puede usar un SDK para obtener acceso MediaStore. SDKs Simplifique la autenticación, integre fácilmente con su entorno de desarrollo y proporcione acceso sencillo a MediaStore los comandos. Para obtener más información, consulte Herramientas para Amazon Web Services.
- Herramientas de AWS para Windows PowerShell: para obtener más información, consulte la Guía del AWS Tools for Windows PowerShell usuario.

Precios de AWS Elemental Elemental Elemental El MediaStore

Al igual que sucede con otros AWS productos de, no hay contratos ni compromisos mínimos de uso MediaStore. Se cobra una tarifa por GB adquirido cuando se incorpora contenido al servicio y una tarifa mensual por GB por el contenido que se almacena en el servicio. Para obtener más información, consulte MediaStore Precios de AWS Elemental Elemental Elemental Elemental Elem

Regiones y puntos de conexión de AWS Elemental Elemental Elemental Elem MediaStore

Para reducir la latencia de datos de las aplicaciones, MediaStore ofrece un punto de conexión regional para realizar solicitudes:

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

Para ver la lista completa de regiones de AWS en las que MediaStore está disponible, consulte MediaStore Puntos de conexión y cuotas de AWS Elemental Elemental Elemental Elemental Elemental Elemental Elemental Elemental Elemental Elemental Elemental Elemental

Configuración de AWS Elemental MediaStore

En esta sección, se explican los pasos necesarios para configurar los usuarios para que accedan a AWS Elemental MediaStore. Para obtener información general e información adicional sobre la administración de identidades y accesos MediaStore, consulteIdentity and Access Management para AWS Elemental MediaStore.

Para empezar a usar AWS Elemental MediaStore, complete los siguientes pasos.

Temas

Inscríbase en una Cuenta de AWS

Si no tiene uno Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirse a una Cuenta de AWS

1. Abrir https://portal.aws.amazon.com/billing/registro.
2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro consiste en recibir una llamada telefónica o mensaje de texto e indicar un código de verificación en el teclado del teléfono.

   Cuando te registras en un Cuenta de AWS, Usuario raíz de la cuenta de AWSse crea un. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar tareas que requieren acceso de usuario raíz.

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. En cualquier momento, puede ver la actividad de su cuenta actual y administrarla accediendo a https://aws.amazon.com/y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión AWS Management Consolecomo propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Iniciar sesión como usuario raíz en la Guía del usuario de AWS Sign-In .
2. Active la autenticación multifactor (MFA) para el usuario raíz.

   Para obtener instrucciones, consulte Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz (consola) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

   Consulte las instrucciones en Activar AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center .
2. En IAM Identity Center, conceda acceso administrativo a un usuario.

   Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la Guía del AWS IAM Identity Center usuario.

Inicio de sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, use la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

  Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte Iniciar sesión en el portal de AWS acceso en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

   Para conocer las instrucciones, consulte Create a permission set en la Guía del usuario de AWS IAM Identity Center .
2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

   Para conocer las instrucciones, consulte Add groups en la Guía del usuario de AWS IAM Identity Center .

Introducción a AWS Elemental MediaStore

Este tutorial de introducción le muestra cómo usar AWS Elemental MediaStore para crear un contenedor y cargar un objeto.

Temas

Paso 1: Acceda a AWS Elemental MediaStore

Una vez que haya configurado su cuenta de AWS y creado los usuarios y roles, inicie sesión en la consola de AWS Elemental MediaStore.

Para acceder a AWS Elemental MediaStore

- Inicie sesión en AWS Management Console y abra la MediaStore consola en https://console.aws.amazon.com/mediastore/.

  > **Note**
  >
  > Puede iniciar sesión con cualquiera de las credenciales de IAM que ha creado para esta cuenta. Para obtener información sobre la creación de credenciales de IAM, consulte Configuración de AWS Elemental MediaStore.

Paso 2: Crear un contenedor

Utiliza contenedores en AWS Elemental MediaStore para almacenar sus carpetas y objetos. Puede utilizar los contenedores para agrupar objetos relacionados del mismo modo en que usa un directorio para agrupar archivos en un sistema de archivos. No se le cobrará por crear contenedores; solo se le cobrará cuando cargue un objeto en un contenedor.

Para crear un contenedor

1. En la página Containers (Contenedores), elija Create container (Crear contenedor).
2. En Container name (Nombre del contenedor), escriba un nombre para el contenedor. Para obtener más información, consulte Reglas para los nombres de contenedor.
3. Elija Crear contenedor. AWS Elemental MediaStore añade el nuevo contenedor a una lista de contenedores. Inicialmente, el estado del contenedor es Creating (Creándose) y luego cambia a Active (Activo).

Paso 3: Cargar un objeto

Puede cargar objetos (de hasta 25 MB cada uno) en un contenedor o en una carpeta dentro de un contenedor. Para cargar un objeto en una carpeta, debe especificar la ruta a la carpeta. Si la carpeta ya existe, AWS Elemental MediaStore almacena el objeto en la carpeta. Si la carpeta no existe, el servicio la crea y, a continuación, almacena el objeto en ella.

> **Note**
>
> Los nombres de archivo de los objetos solo pueden contener letras, números, puntos (.), guiones bajos (_), tildes (~) y guiones (-).

Para cargar un objeto

1. En la página Containers (Contenedores), elija el nombre del contenedor que acaba de crear. Aparecerá la página de detalles del contenedor.
2. Elija Upload object (Cargar objeto).
3. En Target path (Ruta de destino), escriba una ruta para las carpetas. Por ejemplo, `premium/canada`. Si alguna de las carpetas de la ruta aún no existe, AWS Elemental la MediaStore crea automáticamente.
4. En Object (Objeto), elija Browse (Examinar).
5. Vaya a la carpeta correspondiente y, a continuación, elija el objeto que desea cargar.
6. Elija Open (Abrir) y, a continuación, Upload (Cargar).

Paso 4: Obtener acceso a un objeto

Es posible descargar los objetos en un punto de enlace especificado.

1. En la página Containers (Contenedores), elija el nombre del contenedor que tiene el objeto que desea descargar.
2. Si el objeto que desea descargar se encuentra en una subcarpeta, continúe eligiendo los nombres de las carpetas hasta que vea el objeto.
3. Elija el nombre del objeto.
4. En la página de detalles del objeto, elija Download (Descargar).

Contenedores en AWS Elemental MediaStore

Los contenedores se utilizan MediaStore para almacenar sus carpetas y objetos. Los objetos relacionados se pueden agrupar en contenedores del mismo modo en que se usa un directorio para agrupar archivos en un sistema de archivos. No se le cobrará por crear contenedores; solo se le cobrará cuando cargue un objeto en un contenedor. Para obtener más información sobre los cargos, consulte los MediaStoreprecios de AWS Elemental.

Temas

Reglas para los nombres de contenedor

Al elegir un nombre para el contenedor, recuerde lo siguiente:

- El nombre debe ser único dentro del cuenta actual para la región de AWS actual.
- El nombre puede contener letras en mayúsculas y minúsculas, números y guiones bajos (_).
- El nombre debe tener entre 1 y 255 caracteres.
- Los nombres distinguen mayúsculas de minúsculas. Por ejemplo, puede tener un contenedor denominado `myContainer` y una carpeta con el nombre `mycontainer`, ya que esos nombres son únicos.
- No se puede cambiar el nombre de un contenedor una vez creado.

Creación de un contenedor

Puede crear hasta 100 contenedores por cuenta de AWS. Puede crear tantas carpetas como desee, siempre que no estén anidadas más de 10 niveles dentro de un contenedor. Asimismo, puede cargar tantos objetos como desee en cada contenedor.

> **Tip**
>
> También puede crear un contenedor automáticamente mediante una AWS CloudFormation plantilla. La plantilla de AWS CloudFormation administra los datos para cinco acciones de la API: creación de un contenedor, establecimiento del registro de acceso, actualización de la política de contenedor predeterminada, adición de una política de uso compartido de recursos entre orígenes (CORS) y adición de una política de ciclo de vida de objetos. Para obtener más información, consulte la Guía del usuario de AWS CloudFormation.

Para crear un contenedor (consola)

1. Abra la MediaStore consola en https://console.aws.amazon.com/mediastore/.
2. En la página Containers (Contenedores), elija Create container (Crear contenedor).
3. En Container name (Nombre del contenedor), escriba un nombre para el contenedor. Para obtener más información, consulte Reglas para los nombres de contenedor.
4. Elija Crear contenedor. AWS Elemental MediaStore añade el nuevo contenedor a una lista de contenedores. Inicialmente, el estado del contenedor es Creating (Creándose) y luego cambia a Active (Activo).

Para crear un contenedor (AWS CLI)

- En el AWS CLI, utilice el `create-container` comando:

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

En el siguiente ejemplo, se muestra el valor de retorno:

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

```
}
```

Visualización de los detalles de un contenedor

Los detalles de un contenedor incluyen la política de contenedor, el punto de enlace, el ARN y la hora de creación.

Para ver los detalles de un contenedor (consola)

1. Abra la MediaStore consola en. https://console.aws.amazon.com/mediastore/
2. En la página Containers (Contenedores), elija el nombre del contenedor.

   Aparecerá la página de detalles del contenedor. Esta página se divide en dos secciones:

   - La sección Objects (Objetos), que enumera los objetos y las carpetas del contenedor.
   - La sección Container policy (Política del contenedor), que muestra la política basada en recursos que está asociada con este contenedor. Para obtener información sobre las políticas de recursos, consulte Políticas de contenedor.

Para ver los detalles de un contenedor (AWS CLI)

- En AWS CLI, utilice el `describe-container` comando:

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

  En el siguiente ejemplo, se muestra el valor de retorno:

  ```
  {
      "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"
      }
  ```

```
}
```

Visualización de una lista de contenedores

Puede ver una lista de todos los contenedores que están asociados a su cuenta.

Para ver una lista de contenedores (consola)

- Abra la MediaStore consola en https://console.aws.amazon.com/mediastore/.

  Aparece la página Containers (Contenedores), que muestra todos los contenedores que están asociados a la cuenta.

Para ver una lista de contenedores (AWS CLI)

- En AWS CLI, utilice el `list-containers` comando.

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

  En el siguiente ejemplo, se muestra el valor de retorno:

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

Eliminación de un contenedor

Un contenedor únicamente se puede eliminar si no tiene objetos.

Para eliminar un contenedor (consola)

1. Abra la MediaStore consola en https://console.aws.amazon.com/mediastore/.
2. En la página Containers (Contenedores), elija la opción situada a la izquierda del nombre del contenedor.
3. Elija Eliminar.

Para eliminar un contenedor (AWS CLI)

- En AWS CLI, utilice el `delete-container` comando:

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

  Este comando no tiene ningún valor de retorno.

Políticas de AWS Elemental MediaStore

Puede aplicar una o más de estas políticas a su MediaStore contenedor de AWS Elemental:

- Política de contenedores: establece los derechos de acceso a todas las carpetas y objetos del contenedor. MediaStore establece una política predeterminada que permite a los usuarios realizar todas MediaStore las operaciones en el contenedor. Esta política especifica que todas las operaciones deben realizarse a través de HTTPS. Después de crear un contenedor, puede editar la política de contenedor.
- Política de intercambio de recursos entre orígenes (CORS): permite que las aplicaciones web cliente de un dominio interactúen con los recursos de un dominio diferente. MediaStore no establece una política CORS predeterminada.
- Política de métricas: MediaStore permite enviar métricas a Amazon CloudWatch. MediaStore no establece una política de métricas predeterminada.
- Política de ciclo de vida de los objetos: controla el tiempo que permanecen los objetos en un MediaStore contenedor. MediaStoreno establece una política de ciclo de vida de los objetos predeterminada.

Políticas de contenedores en AWS Elemental MediaStore

Cada contenedor tiene una política basada en recursos que rige los derechos de acceso a todas las carpetas y objetos de dicho contenedor. La política predeterminada, que se adjunta automáticamente a todos los contenedores nuevos, permite el acceso a todas las MediaStore operaciones de AWS Elemental en el contenedor. Especifica que este acceso requiere HTTPS para las operaciones. Después de crear un contenedor, puede editar la política que se asocia a dicho contenedor.

También puede utilizar una política de ciclo de vida de objetos que rija la fecha de vencimiento de objetos en un contenedor. Después de que los objetos alcancen la máxima antigüedad especificada, el servicio elimina los objetos del contenedor.

Temas

- Visualización de una política de contenedor
- Edición de una política de contenedor
- Ejemplos de políticas de contenedor

Visualización de una política de contenedor

Puede usar la consola o la AWS CLI para ver la política basada en recursos de un contenedor.

Para ver una política de contenedor (consola)

1. Abra la MediaStore consola en. https://console.aws.amazon.com/mediastore/
2. En la página Containers (Contenedores), elija el nombre del contenedor.

   Aparecerá la página de detalles del contenedor. La política se muestra en la sección Container policy (Política de contenedor).

Para ver una política de contenedor (AWS CLI)

- En AWS CLI, utilice el `get-container-policy` comando:

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

  En el siguiente ejemplo, se muestra el valor de retorno:

  ```
  {
    "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"
            }
  ```

```
        }
      }
    ]
  }
}
```

Edición de una política de contenedor

Puede editar los permisos de la política de contenedor predeterminada, o puede crear una política nueva para sustituirla. Se necesita hasta cinco minutos para que la nueva política surta efecto.

Para editar una política de contenedor (consola)

1. Abra la MediaStore consola en https://console.aws.amazon.com/mediastore/.
2. En la página Containers (Contenedores), elija el nombre del contenedor.
3. Elija Editar política. Para ver ejemplos que ilustran cómo establecer diferentes permisos, consulte the section called “Ejemplos de políticas de contenedor”.
4. Realice los cambios apropiados y elija Guardar.

Para editar una política de contenedor (AWS CLI)

1. Cree un archivo que defina la política del contenedor:

```
{
  "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. En AWS CLI, utilice el `put-container-policy` comando:

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

Este comando no tiene ningún valor de retorno.

Ejemplos de políticas de contenedor

En los siguientes ejemplos se muestran políticas de contenedor creadas para distintos grupos de usuarios.

Temas

Ejemplo de política de contenedor: política predeterminada

Al crear un contenedor, AWS Elemental adjunta MediaStore automáticamente la siguiente política basada en recursos:

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

La política está integrada en el servicio, por lo que no es necesario crearla. Sin embargo, puede editar la política del contenedor si los permisos de la política predeterminada no están alineados con los permisos que quiere usar para el contenedor.

La política predeterminada que se asigna a todos los contenedores nuevos permite el acceso a todas las operaciones de MediaStore en el contenedor. Especifica que este acceso requiere HTTPS para las operaciones.

Ejemplo de política de contenedor: acceso de lectura público a través de HTTPS

Este ejemplo de política permite a los usuarios recuperar un objeto mediante una solicitud HTTPS. Permite acceso de lectura a cualquier persona a través de una conexión SSL/TLS segura: usuarios autenticados y usuarios anónimos (los usuarios que no han iniciado sesión). La instrucción se denomina `PublicReadOverHttps`. Permite el acceso a las operaciones `GetObject` y `DescribeObject` en cualquier objeto (tal como especifica el carácter * al final de la ruta de recurso). Especifica que este acceso requiere HTTPS para las operaciones:

```
{
  "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"
```

```
        }
      }
    }
  ]
}
```

Ejemplo de política de contenedor: acceso de lectura público a través de HTTP o HTTPS

Este ejemplo de política permite el acceso a las operaciones `GetObject` y `DescribeObject` en cualquier objeto (tal como especifica el carácter * al final de la ruta de recurso). Permite acceso de lectura a todo el mundo, incluidos todos los usuarios autenticados y los usuarios anónimos (los usuarios que no han iniciado sesión):

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

Ejemplo de política de contenedor: acceso de lectura entre cuentas con HTTP habilitado

Esta política permite a los usuarios recuperar un objeto mediante una solicitud HTTP. Concede este acceso a los usuarios autenticados con acceso entre cuentas. No es necesario que el objeto esté alojado en un servidor con un certificado SSL/TLS:

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

Ejemplo de política de contenedor: acceso de lectura entre cuentas a través de HTTPS

Esta política de ejemplo permite el acceso a las operaciones `GetObject` y `DescribeObject` en cualquier objeto (tal como especifica el carácter * al final de la ruta de recurso) que pertenezca al usuario raíz del <otro número de cuenta> especificado. Especifica que este acceso requiere HTTPS para las operaciones:

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

```
  ]
}
```

Ejemplo de política de contenedor: acceso de lectura entre cuentas a un rol

La política de ejemplo permite el acceso a las operaciones `GetObject` y `DescribeObject` en cualquier objeto (tal como especifica el carácter * al final de la ruta de recurso) que pertenezca al <número de la cuenta propietaria>. Concede este acceso a los usuarios del <otro número de cuenta> si esa cuenta ha asumido la función que se ha especificado en <nombre de función>:

```
{
  "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>/*",
    }
  ]
}
```

Ejemplo de política de contenedor: acceso completo entre cuentas a un rol

Este ejemplo de política permite acceso entre cuentas para actualizar cualquier objeto de la cuenta, siempre y cuando el usuario haya iniciado sesión sobre HTTP. También permite el acceso entre cuentas para eliminar, descargar y describir objetos a través de HTTP o HTTPS en una cuenta que ha asumido el rol especificado:

- La primera instrucción es `CrossAccountRolePostOverHttps`. Permite el acceso a la operación `PutObject` en cualquier objeto y permite este acceso a los usuarios de la cuenta especificada si esta ha asumido la función especificada en <nombre de función>. Especifica que este acceso requiere HTTPS para la operación (esta condición se debe incluir siempre al proporcionar acceso a `PutObject`).

  En otras palabras, cualquier entidad principal que tenga acceso entre cuentas puede obtener acceso a `PutObject`, pero solo por medio de HTTPS.

- La segunda instrucción es `CrossAccountFullAccessExceptPost`. Permite el acceso a todas las operaciones, excepto a `PutObject`, en cualquier objeto. Concede este acceso a los usuarios de la cuenta especificada si esa cuenta ha asumido la función que se ha especificado en <nombre de función>. Este acceso no requiere HTTPS para las operaciones.

  En otras palabras, cualquier cuenta que tenga acceso entre cuentas puede obtener acceso a `DeleteObject`, `GetObject` y así sucesivamente (pero no a `PutObject`), y puede hacerlo mediante HTTP o HTTPS.

  Si no excluye `PutObject` de la segunda instrucción, la instrucción no será válida (ya que si incluye `PutObject`, debe establecer explícitamente HTTPS como condición).

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

Ejemplo de política de contenedor: acceso restringido a direcciones IP específicas

Este ejemplo de política permite el acceso a todas las MediaStore operaciones de AWS Elemental en los objetos del contenedor especificado. Sin embargo, la solicitud debe proceder del rango de direcciones IP especificado en la condición.

La condición de esta declaración identifica el rango 198.51.100.* de direcciones IP del Protocolo de Internet de la versión 4 (IPv4) permitidas, con una excepción: 198.51.100.188.

El bloque `Condition` utiliza las condiciones `IpAddress` y `NotIpAddress`, y la clave de condición `aws:SourceIp`, que es una clave de condición general de AWS. Los `aws:sourceIp` IPv4 valores utilizan la notación CIDR estándar. Para obtener más información, consulte Operadores de condición de dirección IP en la guía del usuario de IAM.

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

Políticas de uso compartido de recursos entre orígenes (CORS) en AWS Elemental MediaStore

El uso compartido de recursos entre orígenes (CORS) define una manera para que las aplicaciones web de los clientes cargadas en un dominio interactúen con los recursos de un dominio diferente. Con el soporte de CORS en AWS Elemental MediaStore, puede crear aplicaciones web sofisticadas para el lado del cliente MediaStore y permitir el acceso de origen cruzado a sus recursos de forma selectiva. MediaStore

> **Note**
>
> Si utiliza Amazon CloudFront para distribuir contenido desde un contenedor que tiene una política de CORS, asegúrese de configurar la distribución de AWS Elemental MediaStore (incluido el paso de editar el comportamiento de la caché para configurar CORS).

En esta sección, se proporciona información general acerca del CORS. En los subtemas se describe cómo puede habilitar CORS mediante la MediaStore consola de AWS Elemental o mediante programación mediante la API MediaStore REST y AWS. SDKs

Temas

Escenarios de casos de uso de CORS

A continuación, se muestran ejemplos de casos para el uso del CORS:

- Escenario 1: Suponga que está distribuyendo vídeo en streaming en directo en un MediaStore contenedor de AWS Elemental denominado LiveVideo. Los usuarios cargan el punto de enlace del

manifiesto de vídeo `http://livevideo.mediastore.ap-southeast-2.amazonaws.com` desde un origen específico, como `www.example.com`. Desea utilizar un reproductor de JavaScript vídeo para acceder a los vídeos que se originan en este contenedor a través de solicitudes AND no autenticadas`GET`. `PUT` Por lo general, un navegador JavaScript bloquearía el acceso a esas solicitudes, pero puedes establecer una política CORS en tu contenedor para habilitar explícitamente estas solicitudes. `www.example.com`

- Escenario 2: supongamos que quiere alojar la misma transmisión en directo que en el escenario 1 desde su MediaStore contenedor, pero quiere permitir solicitudes de cualquier origen. Puede configurar una política del CORS que especifique el asterisco (*) para los orígenes permitidos, de modo que las solicitudes procedentes de cualquier origen puedan obtener acceso al vídeo.

Agregar una política de CORS a un contenedor

En esta sección se explica cómo añadir una configuración de uso compartido de recursos entre orígenes (CORS) a un contenedor de AWS Elemental MediaStore . CORS permite que las aplicaciones web clientes cargadas en un dominio puedan interactuar con los recursos de otro dominio.

Para configurar un contenedor para permitir las solicitudes entre orígenes, debe añadir una política del CORS al contenedor. Una política del CORS define las reglas que identifican los orígenes desde los que permitirá el acceso al contenedor, las operaciones (métodos HTTP) permitidas para cada origen y otro tipo de información específica de cada operación.

Cuando se añade una política del CORS al contenedor, las políticas del contenedor (que rigen los derechos de acceso al contenedor) seguirán aplicándose.

Para añadir política del CORS (Consola)

1. Abra la MediaStore consola en. https://console.aws.amazon.com/mediastore/
2. En la página Containers (Contenedores), elija el nombre del contenedor para el que desea crear la política del CORS.

   Aparecerá la página de detalles del contenedor.
3. En la sección Container CORS policy (Política del CORS del contenedor), elija Create CORS policy (Crear política del CORS).
4. Inserte la política en formato JSON y, a continuación, elija Save (Guardar).

Para añadir política del CORS (AWS CLI)

1. Cree un archivo que defina la política del CORS:

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

2. En AWS CLI, utilice el `put-cors-policy` comando.

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

   Este comando no tiene ningún valor de retorno.

Visualización de una política de CORS

El uso compartido de recursos entre orígenes (CORS) define una manera para que las aplicaciones web de los clientes cargadas en un dominio interactúen con los recursos de un dominio diferente.

Para ver una política del CORS (consola)

1. Abra la MediaStore consola en https://console.aws.amazon.com/mediastore/.
2. En la página Containers (Contenedores), elija el nombre del contenedor cuya política del CORS desea ver.

   Aparece la página de detalles del contenedor, con la política del CORS en la sección Container CORS policy (Política del CORS del contenedor),

Para ver una política del CORS (AWS CLI)

- En AWS CLI, utilice el `get-cors-policy` comando:

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

En el siguiente ejemplo, se muestra el valor de retorno:

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

Edición de una política de CORS

El uso compartido de recursos entre orígenes (CORS) define una manera para que las aplicaciones web de los clientes cargadas en un dominio interactúen con los recursos de un dominio diferente.

Para editar una política del CORS (consola)

1. Abra la MediaStore consola en https://console.aws.amazon.com/mediastore/.
2. En la página Containers (Contenedores), elija el nombre del contenedor cuya política del CORS desea editar.

   Aparecerá la página de detalles del contenedor.
3. En la sección Container CORS policy (Política del CORS del contenedor), elija Edit CORS policy (Editar política del CORS).

4. Realice los cambios en la política y, a continuación, elija Save (Guardar).

Para editar una política del CORS (AWS CLI)

1. Cree un archivo que defina la política del CORS actualizada:

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

2. En AWS CLI, utilice el `put-cors-policy` comando.

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

Este comando no tiene ningún valor de retorno.

Eliminación de una política de CORS

El uso compartido de recursos entre orígenes (CORS) define una manera para que las aplicaciones web de los clientes cargadas en un dominio interactúen con los recursos de un dominio diferente. La eliminación de la política del CORS de un contenedor elimina los permisos para las solicitudes entre orígenes.

Para eliminar una política del CORS (consola)

1. Abra la MediaStore consola en https://console.aws.amazon.com/mediastore/.

2. En la página Containers (Contenedores), elija el nombre del contenedor cuya política del CORS desea eliminar.

   Aparecerá la página de detalles del contenedor.

3. En la sección Container CORS policy (Política del CORS del contenedor), elija Delete CORS policy (Eliminar política del CORS).
4. Elija Continue (Continuar) para confirmar y, a continuación, elija Save (Guardar).

Para eliminar una política del CORS (AWS CLI)

- En AWS CLI, utilice el `delete-cors-policy` comando:

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

  Este comando no tiene ningún valor de retorno.

Solución de problemas de CORS

Si detecta un comportamiento inesperado al obtener acceso a un contenedor que tiene una política del CORS, siga estos pasos para solucionar el problema.

1. Compruebe que la política del CORS está asociada al contenedor.

   Para obtener instrucciones, consulte the section called “Visualización de una política de CORS”.

2. Capture la solicitud y la respuesta completas con la herramienta que desee (como, por ejemplo, la consola para desarrolladores del navegador). Compruebe que la política del CORS asociada al contenedor incluye al menos una regla del CORS que coincida con los datos de la solicitud, tal y como se indica a continuación:

   a. Compruebe que la solicitud tiene un encabezado `Origin`.

      Si falta el encabezado, AWS Elemental MediaStore no trata la solicitud como una solicitud de origen cruzado y no devuelve los encabezados de respuesta de CORS en la respuesta.

   b. Compruebe que el encabezado `Origin` de la solicitud coincide al menos con uno de los elementos `AllowedOrigins` de la regla `CORSRule` específica.

Los valores de esquema, host y puerto del encabezado de solicitud `Origin` deben coincidir con el elemento `AllowedOrigins` de la regla `CORSRule`. Por ejemplo, si establece que la regla `CORSRule` permita el origen `http://www.example.com`, los orígenes `https://www.example.com` y `http://www.example.com:80` de la solicitud no coinciden con el origen permitido en la configuración.

c. Compruebe que el método de la solicitud (o el método especificado en `Access-Control-Request-Method` en el caso de una solicitud de comprobación preliminar) sea uno de los elementos `AllowedMethods` de la misma regla `CORSRule`.

d. En el caso de una solicitud de comprobación preliminar, si la solicitud incluye un encabezado `Access-Control-Request-Headers`, verifique que la regla `CORSRule` incluya las entradas `AllowedHeaders` para cada valor en el encabezado `Access-Control-Request-Headers`.

Ejemplos de políticas de CORS

En los siguientes ejemplos, se muestran políticas del uso compartido de recursos entre orígenes (CORS).

Temas

- Ejemplo de política de CORS: acceso de lectura para cualquier dominio
- Ejemplo de política de CORS: acceso de lectura para un dominio específico

Ejemplo de política de CORS: acceso de lectura para cualquier dominio

La siguiente política permite que una página web de cualquier dominio recupere contenido de su MediaStore contenedor de AWS Elemental. La solicitud incluye todos los encabezados HTTP del dominio de origen y el servicio responde únicamente a las solicitudes HTTP GET y HTTP HEAD procedentes del dominio de origen. Los resultados se almacenan en caché durante 3 000 segundos antes de que se entregue un conjunto de resultados nuevo.

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

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

Ejemplo de política de CORS: acceso de lectura para un dominio específico

La siguiente política permite que una página web `https://www.example.com` recupere contenido de su MediaStore contenedor de AWS Elemental. La solicitud incluye todos los encabezados HTTP de `https://www.example.com` y el servicio responde únicamente a las solicitudes HTTP GET y HTTP HEAD procedentes de `https://www.example.com`. Los resultados se almacenan en caché durante 3 000 segundos antes de que se entregue un conjunto de resultados nuevo.

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

Políticas del ciclo de vida de los objetos en AWS Elemental MediaStore

Para cada contenedor, puede crear una política de ciclo de vida de objetos que rija el tiempo que los objetos deben almacenarse en el contenedor. Cuando los objetos alcanzan la antigüedad máxima

que especifique, AWS Elemental MediaStore los elimina. Puede eliminar objetos cuando ya no sean necesarios para ahorrar los costos de almacenamiento.

También puede especificar qué objetos se MediaStore deben mover a la clase de almacenamiento de acceso poco frecuente (IA) una vez que hayan alcanzado cierta antigüedad. Los objetos que se almacenan en la clase de almacenamiento IA tienen tasas de almacenamiento y recuperación diferentes a los objetos almacenados en la clase de almacenamiento estándar. Para obtener más información, consulte MediaStore Precios.

Una política de ciclo de vida de objetos contiene reglas que determinan la vida útil de objetos por subcarpeta. (No puede asignar una política de ciclo de vida de objetos a objetos individuales). Puede asociar una única política del ciclo de vida de objetos a un contenedor, pero puede añadir hasta 10 reglas a cada política de ciclo de vida de objetos. Para obtener más información, consulte Componentes de una política de ciclo de vida de objetos.

Temas

Componentes de una política de ciclo de vida de objetos

Las políticas del ciclo de vida de los objetos rigen el tiempo que permanecen los objetos en un MediaStore contenedor de AWS Elemental. Cada política de ciclo de vida de objetos se compone de una o varias reglas, que determinan la vida útil de objetos. Una regla puede aplicarse a una carpeta, a varias carpetas o al contenedor completo.

Puede asociar una política de ciclo de vida de objetos a un contenedor, y cada uno de los objetos de la política de ciclo de vida puede contener hasta 10 reglas. No se puede asignar una política de ciclo de vida de objetos a un objeto individual.

Reglas en una política de ciclo de vida de objetos

Puede crear tres tipos de reglas:

- Datos transitorios
- Eliminar objeto
- Transición de ciclo de vida

Datos transitorios

Una regla de datos transitorios establece que los objetos venzan en cuestión de segundos. Este tipo de regla solo se aplica a los objetos que se agregan al contenedor una vez que la política entra en vigor. Se necesitan hasta 20 minutos MediaStore para aplicar la nueva política al contenedor.

Un ejemplo de una regla para datos transitorios tiene este aspecto:

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

Las reglas de datos transitorios tienen tres partes:

- `path`: siempre se establece en `wildcard`. Puede utilizar esta parte para definir qué objetos desea eliminar. Puede utilizar uno o varios comodines, representados por un asterisco (*). Cada comodín representa cualquier combinación de cero o más caracteres. Por ejemplo, `"path": [ {"wildcard": "Football/index*.m3u8"} ],` se aplica a todos los archivos de la carpeta `Football` que coinciden con el patrón `index*.m3u8` (como index.m3u8, index1.m3us8 e index123456.m3u8). Puede incluir hasta 10 rutas en una sola regla.
- `seconds_since_create`: siempre se establece en `numeric`. Puede especificar un valor de 1 a 300 segundos. También puede establecer el operador en mayor que (**>**) o mayor o igual que (**>=**).
- `action`: siempre se establece en `EXPIRE`.

Para las reglas de datos transitorios (los objetos vencen en cuestión de segundos), no hay ningún retraso entre el momento en que un objeto vence y la eliminación del mismo.

> **Note**
>
> Los objetos sujetos a una regla de datos transitorios no se incluyen en una respuesta `list-items`. Además, los objetos que caducan debido a una regla de datos transitorios no emiten ningún CloudWatch evento cuando caducan.

Eliminar objeto

Una regla de eliminación de objetos establece que los objetos venzan en cuestión de días. Este tipo de regla se aplica a todos los objetos del contenedor, incluso si se añadieron al contenedor antes de que se creara la política. La aplicación de la nueva política tarda hasta 20 minutos, pero los objetos pueden tardar hasta 24 horas en salir del contenedor. MediaStore

Un ejemplo de dos reglas para eliminar objetos tiene este aspecto:

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

Las reglas de eliminación de objetos tienen tres partes:

- `path`: se establece en `prefix` o `wildcard`. No se puede utilizar `prefix` y `wildcard` en la misma regla. Si desea utilizar ambos, debe crear una regla para `prefix` y una regla distinta para `wildcard`, como se muestra en el ejemplo anterior.

  - `prefix`: establezca la ruta de acceso en `prefix` si desea eliminar todos los objetos dentro de una determinada carpeta. Si el parámetro está vacío (`"path": [ { "prefix": "" } ],`), el destino son todos los objetos que se almacenan en cualquier lugar del contenedor actual. Puede incluir hasta 10 rutas `prefix` en una sola regla.
  - `wildcard`: establezca la ruta de acceso en `wildcard` si desea eliminar objetos específicos basados en el nombre de archivo y/o tipo de archivo. Puede utilizar uno o varios comodines, representados por un asterisco (*). Cada comodín representa cualquier combinación de cero o más caracteres. Por ejemplo, `"path": [ {"wildcard": "Football/*.ts"} ],` se aplica a todos los archivos de la carpeta `Football` que coincidan con el patrón `*.ts` (como nombreArchivo.ts, nombreArchivo1.ts y nombreArchivo123456.ts). Puede incluir hasta 10 rutas `wildcard` en una sola regla.
- `days_since_create`: siempre se establece en `numeric`. Puede especificar un valor de 1 a 36 500 días. También puede establecer el operador en mayor que (**>**) o mayor o igual que (**>=**).
- `action`: siempre se establece en `EXPIRE`.

Para las reglas de eliminación de objetos (los objetos vencen en cuestión de días), es posible que haya un pequeño retardo desde que vence un objeto hasta que se elimina. Sin embargo, los cambios en la facturación se producen tan pronto como caduca el objeto. Por ejemplo, si una regla de ciclo de vida especifica 10 `days_since_create`, no se factura el objeto en la cuenta después de que el objeto tenga 10 días de antigüedad, incluso si el objeto aún no se ha eliminado.

Transición de ciclo de vida

Una regla de transición de ciclo de vida establece que los objetos se moverán a la clase de almacenamiento de acceso infrecuente (IA) después de que alcancen una cierta antigüedad, medida en días. Los objetos que se almacenan en la clase de almacenamiento IA tienen tasas de almacenamiento y recuperación diferentes a los objetos almacenados en la clase de almacenamiento estándar. Para más información, consulte Precios de MediaStore.

Una vez que un objeto se ha movido a la clase de almacenamiento IA, no puede volver a moverlo a la clase de almacenamiento estándar.

La regla de transición del ciclo de vida se aplica a todos los objetos del contenedor, incluso si se añadieron al contenedor antes de que se creara la política. La nueva política tarda hasta 20 minutos en aplicarse, pero los objetos pueden tardar hasta 24 horas en salir del contenedor. MediaStore

Un ejemplo de una regla de transición de ciclo de vida es así:

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

Las reglas de transición de ciclo de vida tienen tres partes:

- `path`: se establece en `prefix` o `wildcard`. No se puede utilizar `prefix` y `wildcard` en la misma regla. Si desea utilizar ambos, debe crear una regla para `prefix` y otra regla independiente para `wildcard`.
  - `prefix`: establecer la ruta de acceso a `prefix` si desea la transición de todos los objetos dentro de una carpeta particular a la clase de almacenamiento IA. Si el parámetro está vacío (`"path": [ { "prefix": "" } ],`), el destino son todos los objetos que se guardan en cualquier lugar del contenedor actual. Puede incluir hasta 10 rutas `prefix` en una sola regla.
  - `wildcard`: se establece la ruta de acceso a `wildcard` si desea la transición de objetos específicos a la clase de almacenamiento IA basado en el nombre de archivo y/o tipo de archivo. Puede utilizar uno o varios comodines, representados por un asterisco (*). Cada comodín representa cualquier combinación de cero o más caracteres. Por ejemplo, `"path": [ {"wildcard": "Football/*.ts"} ],` se aplica a todos los archivos de la carpeta `Football` que coincidan con el patrón `*.ts` (como nombreArchivo.ts, nombreArchivo1.ts y nombreArchivo123456.ts). Puede incluir hasta 10 rutas `wildcard` en una sola regla.
- `days_since_create`: siempre se establece en `"numeric": [">=" , 30]`.
- `action`: siempre se establece en `ARCHIVE`.

Ejemplo

Supongamos que un contenedor denominado `LiveEvents` tenga cuatro subcarpetas: `Football`, `Baseball`, `Basketball` y `AwardsShow`. La política de ciclo de vida de objetos asignada a la carpeta `LiveEvents` puede tener un aspecto similar al siguiente:

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

La política anterior especifica los elementos siguientes:

- La primera regla indica MediaStore a AWS Elemental que elimine los objetos que estén almacenados en la `LiveEvents/Football` carpeta y en la `LiveEvents/Baseball` carpeta cuando tengan más de 28 días.
- La segunda regla indica al servicio que elimine los objetos almacenados en la carpeta `LiveEvents/AwardsShow` cuando tengan una antigüedad de 15 días o más.
- La tercera regla indica al servicio eliminar objetos almacenados en cualquier lugar del contenedor `LiveEvents` cuando tengan una antigüedad de 40 días. Esta regla se aplica a los objetos almacenados directamente en el contenedor `LiveEvents`, así como a los objetos almacenados en cualquiera de las cuatro subcarpetas del contenedor.
- La cuarta regla indica al servicio que elimine los objetos de la carpeta `Football` que coincidan con el patrón `*.ts` cuando tengan más de 20 días.
- La quinta regla indica al servicio que elimine los objetos de la `Football` carpeta que coincidan con el patrón `index*.m3u8` cuando tengan más de 15 segundos. MediaStore elimina estos archivos 16 segundos después de colocarlos en el contenedor.

- La sexta regla indica al servicio que mueva los objetos de la carpeta `Program` a la clase de almacenamiento IA después de que tengan 30 días de antigüedad.

Para obtener más ejemplos de políticas de ciclo de vida de objetos, consulte Ejemplo de políticas de ciclo de vida de objetos.

Agregar una política de ciclo de vida de objetos a un contenedor

Una política de ciclo de vida de objetos le permite especificar el tiempo que se almacenan sus objetos en un contenedor. Usted establece una fecha de caducidad y, después de esa fecha, AWS Elemental MediaStore elimina los objetos. El servicio tarda hasta 20 minutos en aplicar la nueva política al contenedor.

Para obtener información acerca de cómo crear una política de ciclo de vida, consulte Componentes de una política de ciclo de vida de objetos.

> **Note**
>
> Para las reglas de eliminación de objetos (los objetos vencen en cuestión de días), es posible que haya un pequeño retardo desde que vence un objeto hasta que se elimina. Sin embargo, los cambios en la facturación se producen tan pronto como caduca el objeto. Por ejemplo, si una regla de ciclo de vida especifica 10 `days_since_create`, no se factura el objeto en la cuenta después de que el objeto tenga 10 días de antigüedad, incluso si el objeto aún no se ha eliminado.

Para añadir una política de ciclo de vida de objetos (consola)

1. Abra la MediaStore consola en. https://console.aws.amazon.com/mediastore/
2. En la página Containers (Contenedores), elija el nombre del contenedor para el que desea crear la política de ciclo de vida de objetos.

   Aparecerá la página de detalles del contenedor.
3. En la sección Object lifecycle policy (Política de ciclo de vida de objeto) elija Create object lifecycle policy (Crear política de ciclo de vida de objetos).
4. Inserte la política en formato JSON y, a continuación, elija Save (Guardar).

Para añadir una política de ciclo de vida de objetos (AWS CLI)

1. Cree un archivo que defina la política de ciclo de vida de objetos:

```
{
    "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. En AWS CLI, utilice el `put-lifecycle-policy` comando:

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

Este comando no tiene ningún valor de retorno. El servicio asocia la política especificada al contenedor.

Visualización de una política de ciclo de vida de objetos

Una política de ciclo de vida de objetos especifica cuánto tiempo deben almacenarse los objetos en un contenedor.

Para ver una política de ciclo de vida de objetos (consola)

1. Abra la MediaStore consola en https://console.aws.amazon.com/mediastore/.
2. En la página Containers (Contenedores), elija el nombre del contenedor cuya política de ciclo de vida de objetos desea ver.

   Aparecerá la página de detalles del contenedor, con la política de ciclo de vida de objetos en la sección Object lifecycle policy (Política de ciclo de vida de objetos).

Para ver una política de ciclo de vida de objetos (AWS CLI)

- En AWS CLI, utilice el `get-lifecycle-policy` comando:

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

En el siguiente ejemplo, se muestra el valor de retorno:

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

Edición de una política de ciclo de vida de objetos

No se puede editar una política de ciclo de vida de objetos existente. Sin embargo, puede cambiar una política existente cargando una política de sustitución. El servicio tarda hasta 20 minutos en aplicar la política actualizada al contenedor.

Para editar una política de ciclo de vida de objetos (consola)

1. Abra la MediaStore consola en https://console.aws.amazon.com/mediastore/.
2. En la página Containers (Contenedores), elija el nombre del contenedor cuya política de ciclo de vida de objetos desea editar.

   Aparecerá la página de detalles del contenedor.
3. En la sección Object lifecycle policy (Política de ciclo de vida de objetos) elija Edit object lifecycle policy (Editar política de ciclo de vida de objetos).
4. Realice los cambios en la política y, a continuación, elija Save (Guardar).

Para editar una política de ciclo de vida de objetos (AWS CLI)

1. Cree un archivo que defina la política de ciclo de vida de objetos actualizada:

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

2. En AWS CLI, utilice el `put-lifecycle-policy` comando:

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

Este comando no tiene ningún valor de retorno. El servicio asocia la política especificada al contenedor, sustituyendo la política anterior.

Eliminación de una política de ciclo de vida de objetos

Cuando elimina una política de ciclo de vida de un objeto, el servicio tarda hasta 20 minutos en aplicar el cambio al contenedor.

Para eliminar una política de ciclo de vida de objetos (consola)

1. Abra la MediaStore consola en https://console.aws.amazon.com/mediastore/.
2. En la página Containers (Contenedores), elija el nombre del contenedor cuya política de ciclo de vida de objetos desea eliminar.

   Aparecerá la página de detalles del contenedor.
3. En la sección Object lifecycle policy (Política de ciclo de vida de objetos) elija Delete lifecycle policy (Eliminar política de ciclo de vida de objetos).
4. Elija Continue (Continuar) para confirmar y, a continuación, elija Save (Guardar).

Para eliminar una política de ciclo de vida de objetos (AWS CLI)

- En AWS CLI, utilice el `delete-lifecycle-policy` comando:

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

Este comando no tiene ningún valor de retorno.

Ejemplo de políticas de ciclo de vida de objetos

En los ejemplos siguientes se muestran las políticas de ciclo de vida de objetos.

Temas

Ejemplo de política de ciclo de vida de objeto: caduca en cuestión de segundos

La siguiente política especifica que MediaStore se eliminarán los objetos que cumplan todos los criterios siguientes:

- El objeto se agrega al contenedor una vez que la política entra en vigor.
- El objeto se almacena en la carpeta `Football`.
- El objeto tiene una extensión de archivo de `m3u8`.
- El objeto ha estado en el contenedor durante más de 20 segundos.

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

Ejemplo de política de ciclo de vida de objeto: caduca en días

La siguiente política especifica que se MediaStore eliminen los objetos que cumplan todos los criterios siguientes:

- El objeto se almacena en la carpeta `Program`
- El objeto tiene una extensión de archivo de `ts`
- El objeto ha estado en el contenedor durante más de 5 días

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

Ejemplo de política de ciclo de vida de objeto: transición a clase de almacenamiento de acceso infrecuente

La siguiente política especifica que los objetos MediaStore se mueven a la clase de almacenamiento de acceso poco frecuente (IA) cuando tienen 30 días de antigüedad. Los objetos que se almacenan en la clase de almacenamiento IA tienen tasas de almacenamiento y recuperación diferentes a los objetos almacenados en la clase de almacenamiento estándar.

El campo `days_since_create` debe establecerse en `"numeric": [">=" ,30]`.

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

```
                ]
            },
            "action": "ARCHIVE"
        }
    ]
}
```

Ejemplo de política de ciclo de vida de objeto: múltiples reglas

La siguiente política especifica que MediaStore hace lo siguiente:

- Mover los objetos almacenados en la carpeta `AwardsShow` a la clase de almacenamiento de acceso infrecuente (IA) después de 30 días
- Eliminar objetos que tienen una extensión de archivo de `m3u8` y se almacenan en la carpeta `Football` después de 20 segundos
- Eliminar objetos almacenados en la carpeta `April` después de 10 días
- Eliminar objetos que tienen una extensión de archivo de `ts` y se almacenan en la carpeta `Program` después de 5 días

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

Ejemplo de política de ciclo de vida de objeto: contenedor vacío

La siguiente política de ciclo de vida de los objetos especifica MediaStore que se eliminarán todos los objetos del contenedor, incluidas las carpetas y subcarpetas, un día después de haberlos agregado al contenedor. Si el contenedor contiene objetos antes de que se aplique esta política, MediaStore los eliminará un día después de que la política entre en vigor. El servicio tarda hasta 20 minutos en aplicar la nueva política al contenedor.

```
{
    "rules": [
        {
            "definition": {
                "path": [
                    {"wildcard": "*"}
                ],
```

```
                "days_since_create": [
                    {"numeric": [ ">=", 1 ]}
                ]
            },
            "action": "EXPIRE"
        }
    ]
}
```

Políticas de métricas en AWS Elemental MediaStore

Para cada contenedor, puede añadir una política de métricas que permita a AWS Elemental MediaStore enviar métricas a Amazon CloudWatch. Se necesitan hasta 20 minutos para que la nueva política surta efecto. Para obtener una descripción de cada MediaStore métrica, consulteMediaStore métricas.

Una política de métricas contiene lo siguiente:

- Un valor para habilitar o deshabilitar las métricas en el nivel de contenedor.
- Entre cero y cinco reglas que habilitan métricas en el nivel de objeto. Si la política contiene reglas, cada regla debe incluir lo siguiente:
  - Un grupo de objetos que define los objetos que se van a incluir en el grupo. La definición puede ser una ruta de acceso o un nombre de archivo, pero no puede tener más de 900 caracteres. Los caracteres válidos son: a-z, A-Z, 0-9, _ (guión bajo), = (igual), : (dos puntos), . (punto), - (guión), ~ (tilde), / (barra diagonal) y * (asterisco). Se admiten caracteres comodín (*).
  - Un nombre de grupo de objetos que permite hacer referencia al grupo de objetos. El nombre no puede tener más de 30 caracteres. Los caracteres válidos son a-z, A-Z, 0-9 y _ (guion bajo).

Si un objeto coincide con varias reglas, CloudWatch muestra un punto de datos para cada regla coincidente. Por ejemplo, si un objeto coincide con dos reglas denominadas `rule1` y`rule2`, CloudWatch muestra dos puntos de datos para estas reglas. El primero tiene una dimensión de `ObjectGroupName=rule1` y el segundo tiene una dimensión de `ObjectGroupName=rule2`.

Temas

- Agregar una política de métricas
- Visualización de una política de métricas
- Edición de una política de métricas

- Políticas de métricas de ejemplo

Agregar una política de métricas

Una política de métricas contiene reglas que determinan qué métricas MediaStore envía AWS Elemental a Amazon CloudWatch. Para obtener ejemplos de políticas de métricas, consulte Políticas de métricas de ejemplo.

Para agregar una política de métricas (consola)

1. Abra la MediaStore consola en. https://console.aws.amazon.com/mediastore/
2. En la página Containers (Contenedores), elija el nombre del contenedor para el que desea agregar una política de métricas.

   Aparecerá la página de detalles del contenedor.
3. En la sección Metric policy (Política de métricas), elija Create metric policy (Crear política de métricas).
4. Inserte la política en formato JSON y, a continuación, elija Save (Guardar).

Visualización de una política de métricas

Puede utilizar la consola o la AWS CLI para ver la política de métricas de un contenedor.

Para ver una política de métrica (consola)

1. Abra la MediaStore consola en https://console.aws.amazon.com/mediastore/.
2. En la página Containers (Contenedores), elija el nombre del contenedor.

   Aparecerá la página de detalles del contenedor. La política se muestra en la sección Metric policy (Política de métricas).

Edición de una política de métricas

Una política de métricas contiene reglas que determinan qué métricas MediaStore envía AWS Elemental a Amazon CloudWatch. Cuando edita una política de métricas existente, la nueva política tarda hasta 20 minutos en entrar en vigor. Para obtener ejemplos de políticas de métricas, consulte Políticas de métricas de ejemplo.

Para editar una política de métricas (consola)

1. Abra la MediaStore consola en. https://console.aws.amazon.com/mediastore/
2. En la página Containers (Contenedores), elija el nombre del contenedor.
3. En la sección Metric policy (Política de métricas) elija Edit metric policy (Editar política de métricas).
4. Realice los cambios apropiados y elija Guardar.

Políticas de métricas de ejemplo

En los siguientes ejemplos se muestran políticas de métricas creadas para distintos casos de uso.

Temas

- Ejemplo de política de métricas: métricas de nivel de contenedor
- Ejemplo de política de métricas: métricas de nivel de ruta
- Ejemplo de política de métricas: métricas de nivel de contenedor y de ruta
- Ejemplo de política de métricas: métricas de nivel de ruta con caracteres comodín
- Ejemplo de política de métricas: métricas de nivel de ruta con reglas solapadas

Ejemplo de política de métricas: métricas de nivel de contenedor

En este ejemplo de política se indica que AWS Elemental MediaStore debe enviar las métricas a Amazon CloudWatch a nivel de contenedor. Por ejemplo, esto incluye la métrica `RequestCount`, que cuenta el número de solicitudes `Put` realizadas al contenedor. También puede establecer este valor en `DISABLED`.

Como esta política no contiene reglas, MediaStore no envía métricas a nivel de ruta. Por ejemplo, no puede ver cuántas solicitudes `Put` se han realizado en una carpeta concreta dentro de este contenedor.

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

Ejemplo de política de métricas: métricas de nivel de ruta

Este ejemplo de política indica que AWS Elemental no MediaStore debe enviar métricas a Amazon a CloudWatch nivel de contenedor. Además, MediaStore debe enviar métricas para los objetos de dos carpetas específicas: `baseball/saturday` y `football/saturday`. Las métricas de las solicitudes de MediaStore son las siguientes:

- Las solicitudes a la `baseball/saturday` carpeta tienen una CloudWatch dimensión de`ObjectGroupName=baseballGroup`.
- Las solicitudes a la carpeta `football/saturday` tienen una dimensión `ObjectGroupName=footballGroup`.

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

Ejemplo de política de métricas: métricas de nivel de contenedor y de ruta

En este ejemplo de política se indica que AWS Elemental MediaStore debe enviar las métricas a Amazon CloudWatch a nivel de contenedor. Además, MediaStore debería enviar las métricas de los objetos de dos carpetas específicas: `baseball/saturday` y`football/saturday`. Las métricas de las solicitudes de MediaStore son las siguientes:

- Las solicitudes a la `baseball/saturday` carpeta tienen una CloudWatch dimensión de`ObjectGroupName=baseballGroup`.
- Las solicitudes a la `football/saturday` carpeta tienen una CloudWatch dimensión`ObjectGroupName=footballGroup`.

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

Ejemplo de política de métricas: métricas de nivel de ruta con caracteres comodín

En este ejemplo de política se indica que AWS Elemental MediaStore debe enviar las métricas a Amazon CloudWatch a nivel de contenedor. Además, también MediaStore debería enviar métricas de los objetos en función de su nombre de archivo. Un carácter comodín indica que los objetos se pueden almacenar en cualquier lugar del contenedor y que pueden tener cualquier nombre de archivo, siempre que termine con una extensión `.m3u8`.

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

Ejemplo de política de métricas: métricas de nivel de ruta con reglas solapadas

En este ejemplo de política se indica que AWS Elemental MediaStore debe enviar las métricas a Amazon CloudWatch a nivel de contenedor. Además, MediaStore debería enviar las métricas de dos carpetas: `sports/football/saturday` y`sports/football`.

Las métricas de MediaStore las solicitudes a la `sports/football/saturday` carpeta tienen una CloudWatch dimensión de`ObjectGroupName=footballGroup1`. Como los objetos almacenados en la carpeta `sports/football` coinciden con ambas reglas, CloudWatch muestra dos puntos

de datos para estos objetos: uno con una dimensión `ObjectGroupName=footballGroup1` y el segundo con una dimensión `ObjectGroupName=footballGroup2`.

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

Carpetas en AWS Elemental MediaStore

Las carpetas son divisiones de un contenedor. Utilícelas para subdividir el contenedor de la misma forma que crea subcarpetas para dividir una carpeta en un sistema de archivos. Se pueden crear hasta 10 niveles de carpetas (sin incluir el propio contenedor).

Las carpetas son opcionales; si lo desea, puede cargar los objetos directamente en un contenedor en lugar de en una carpeta. Sin embargo, las carpetas son una forma sencilla de organizar los objetos.

Para cargar un objeto en una carpeta, debe especificar la ruta a la carpeta. Si la carpeta ya existe, AWS Elemental MediaStore almacena el objeto en la carpeta. Si la carpeta no existe, el servicio la crea y, a continuación, almacena el objeto en ella.

Por ejemplo, supongamos que tiene un contenedor denominado `movies` y que carga el archivo `mlaw.ts` con la ruta `premium/canada`. AWS Elemental MediaStore almacena el objeto en la subcarpeta canada, en la carpeta premium. Si no existe ninguna de las carpetas, el servicio creará las subcarpetas `premium` y `canada`; a continuación, almacenará el objeto en la subcarpeta `canada`. Si solo especifica el contenedor `movies` (sin ninguna ruta), el servicio almacena el objeto directamente en el contenedor.

AWS Elemental elimina MediaStore automáticamente una carpeta al eliminar el último objeto de esa carpeta. El servicio también elimina las carpetas vacías que están por encima de esa carpeta. Por ejemplo, supongamos que tiene una carpeta denominada premium que no contiene ningún archivo, pero que contiene la subcarpeta `canada`. La subcarpeta `canada` contiene un archivo denominado `mlaw.ts`. Si elimina el archivo `mlaw.ts`, el servicio elimina las carpetas `premium` y `canada`. Esta eliminación automática se aplica únicamente a las carpetas. El servicio no elimina contenedores vacíos.

Temas

Reglas para los nombres de carpeta

Al elegir un nombre para la carpeta, recuerde lo siguiente:

- El nombre solo puede contener los siguientes caracteres: letras mayúsculas (A-Z), letras minúsculas (a-z), números (0-9), puntos (.), guiones (-), tildes (~), subrayados (_), signos de igual (=) y de doble punto (:).
- El nombre debe contener al menos un carácter. No se permiten nombres de carpetas vacíos (como `folder1//folder3/`).
- Los nombres distinguen mayúsculas de minúsculas. Por ejemplo, puede tener una carpeta denominada `myFolder` y una carpeta con el nombre `myfolder` en el mismo contenedor o carpeta, ya que esos nombres son únicos.
- El nombre debe ser único solo dentro del contenedor o la carpeta principal. Por ejemplo, puede crear una carpeta con el nombre `myfolder` en dos contenedores diferentes: `movies/myfolder` y `sports/myfolder`.
- El nombre puede tener el mismo nombre que su contenedor principal.
- No se puede cambiar el nombre de la carpeta una vez creada.

Creación de una carpeta

Puede crear carpetas al cargar los objetos. Para cargar un objeto en una carpeta, debe especificar la ruta a la carpeta. Si la carpeta ya existe, AWS Elemental MediaStore almacena el objeto en la carpeta. Si la carpeta no existe, el servicio la crea y, a continuación, almacena el objeto en ella.

Para obtener más información, consulte the section called “Carga de un objeto”.

Eliminación de una carpeta

Solo se pueden eliminar las carpeta que están vacías; no es posible eliminar carpetas que contengan objetos.

AWS Elemental elimina MediaStore automáticamente una carpeta al eliminar el último objeto de esa carpeta. El servicio también elimina las carpetas vacías que están por encima de esa carpeta. Por ejemplo, supongamos que tiene una carpeta denominada `premium` que no contiene ningún archivo, pero que contiene la subcarpeta `canada`. La subcarpeta `canada` contiene un archivo denominado `mlaw.ts`. Si elimina el archivo `mlaw.ts`, el servicio elimina las carpetas `premium` y `canada`. Esta

eliminación automática se aplica únicamente a las carpetas. El servicio no elimina contenedores vacíos.

Para obtener más información, consulte Eliminación de un objeto.

Objetos en AWS Elemental MediaStore

Los MediaStore activos de AWS Elemental se denominan objetos. Puede cargar un objeto en un contenedor o en una carpeta dentro del contenedor.

En MediaStore, puede cargar, descargar y eliminar objetos:

- Cargar: añadir un objeto a un contenedor o a una carpeta. Esto no es lo mismo que crear el objeto. Debe crear sus objetos localmente antes de poder cargarlos allí MediaStore.
- Descargar: copia un objeto desde MediaStore otra ubicación. Esto no elimina el objeto de MediaStore.
- Eliminar: eliminar un objeto de MediaStore definitivamente. Puede eliminar objetos individualmente, o puede añadir una política del ciclo de vida de objeto para eliminar automáticamente objetos dentro de un contenedor después de una duración especificada.

MediaStore acepta todos los tipos de archivos.

Temas

Carga de un objeto

Puede cargar objetos en un contenedor o en una carpeta dentro de un contenedor. Para cargar un objeto en una carpeta, debe especificar la ruta a la carpeta. Si la carpeta ya existe, AWS Elemental MediaStore almacena el objeto en la carpeta. Si la carpeta no existe, el servicio la crea y, a continuación, almacena el objeto en ella. Para obtener más información sobre las carpetas consulte Carpetas en AWS Elemental MediaStore.

Puede usar la MediaStore consola o la AWS CLI para cargar objetos.

MediaStore admite la transferencia fragmentada de objetos, lo que reduce la latencia al permitir que un objeto esté disponible para su descarga mientras se está cargando. Para utilizar esta capacidad,

establezca la disponibilidad de carga del objeto en `streaming`. Puede establecer el valor de este encabezado cuando cargue el objeto mediante la API. Si no especificas este encabezado en tu solicitud, MediaStore asigna el valor predeterminado de `standard` para la disponibilidad de carga del objeto.

Los tamaños de objetos no pueden superar los 25 MB para disponibilidad de carga estándar y los 10 MB para disponibilidad de carga de streaming.

> **Note**
>
> Los nombres de archivo de los objetos solo pueden contener letras, números, puntos (.), guiones bajos (_), tildes (~), guiones (-), signos de igual (=) y dos puntos (:).

Para cargar un objeto (consola)

1. Abre la MediaStore consola en. https://console.aws.amazon.com/mediastore/
2. En la página Containers (Contenedores), elija el nombre del contenedor. Aparecerá el panel de detalles del contenedor.
3. Elija Upload object (Cargar objeto).
4. En Target path (Ruta de destino), escriba una ruta para las carpetas. Por ejemplo, `premium/canada`. Si no existe alguna de las carpetas de la ruta que especifique, el servicio la crea automáticamente.
5. En la sección Object (Objeto), elija Browse (Examinar).
6. Vaya a la carpeta correspondiente y, a continuación, elija el objeto que desea cargar.
7. Elija Open (Abrir) y, a continuación, Upload (Cargar).

   > **Note**
   >
   > Si ya existe un archivo con el mismo nombre en la carpeta seleccionada, el servicio sustituye el archivo original por el archivo cargado.

Para cargar un objeto (AWS CLI)

- En AWS CLI, utilice el `put-object` comando. También puede incluir cualquiera de los siguientes parámetros: `content-type`, `cache-control` (para permitir que el autor de la

llamada controle el comportamiento de la caché del objeto) y `path` (para colocar el objeto en una carpeta dentro del contenedor).

> **Note**
>
> Después de cargar el objeto, no puede editar `content-type`, `cache-control` ni `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
```

En el siguiente ejemplo, se muestra el valor de retorno:

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

Visualización de una lista de objetos

Puede usar la MediaStore consola de AWS Elemental para ver los elementos (objetos y carpetas) almacenados en el nivel superior de un contenedor o en una carpeta. Los elementos almacenados en una subcarpeta del contenedor o la carpeta actual no se mostrarán. Puede utilizarla AWS CLI para ver una lista de objetos y carpetas dentro de un contenedor, independientemente del número de carpetas o subcarpetas que contenga.

Para ver una lista de los objetos de un contenedor específico (consola)

1. Abra la MediaStore consola en. https://console.aws.amazon.com/mediastore/
2. En la página Containers (Contenedores), elija el nombre del contenedor en el que se encuentra la carpeta que desea ver.
3. Elija el nombre de la carpeta en la lista.

Aparecerá una página de detalles, en la que se muestran todas las carpetas y los objetos que contiene la carpeta.

Para ver una lista de los objetos de una carpeta específica (consola)

1. Abra la MediaStore consola en https://console.aws.amazon.com/mediastore/.
2. En la página Containers (Contenedores), elija el nombre del contenedor en el que se encuentra la carpeta que desea ver.

   Aparecerá una página de detalles, en la que se muestran todas las carpetas y los objetos que contiene el contenedor.

Para ver una lista de los objetos y las carpetas de un contenedor específico (AWS CLI)

- En AWS CLI, utilice el `list-items` comando:

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

En el siguiente ejemplo, se muestra el valor de retorno:

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

> **Note**
>
> Los objetos sujetos a una regla `seconds_since_create` no se incluyen en una respuesta `list-items`.

Para ver una lista de los objetos y las carpetas de una carpeta específica (AWS CLI)

- En el AWS CLI, utilice el `list-items` comando, con el nombre de la carpeta especificada al final de la solicitud:

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

En el siguiente ejemplo, se muestra el valor de retorno:

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

> **Note**
>
> Los objetos sujetos a una regla `seconds_since_create` no se incluyen en una respuesta `list-items`.

Visualización de los detalles de un objeto

Después de cargar un objeto, AWS Elemental MediaStore almacena detalles como la fecha de modificación, la longitud del contenido ETag (etiqueta de entidad) y el tipo de contenido. Para obtener información sobre cómo se utilizan los metadatos de un objeto, consulte Interacción de MediaStore con cachés HTTP.

Para ver los detalles de un objeto (consola)

1. Abra la MediaStore consola en https://console.aws.amazon.com/mediastore/.
2. En la página Containers (Contenedores), elija el nombre del contenedor en el que se encuentra el objeto que desea ver.
3. Si el objeto que desea ver se encuentra en una carpeta, continúe eligiendo los nombres de las carpetas hasta que vea el objeto.
4. Elija el nombre del objeto.

   Aparecerá una página de detalles, en la que se muestra la información sobre el objeto.

Para ver los detalles de un objeto (AWS CLI)

- En AWS CLI, utilice el `describe-object` comando:

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

En el siguiente ejemplo, se muestra el valor de retorno:

```
{
    "ContentType": "image/jpeg",
    "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
```

```
    "ContentLength": "2307346",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
}
```

Descarga de un objeto

Puede utilizar la consola para descargar un objeto. Puede utilizarla AWS CLI para descargar un objeto o solo una parte de él.

Para descargar un objeto (consola)

1. Abra la MediaStore consola en https://console.aws.amazon.com/mediastore/.
2. En la página Containers (Contenedores), elija el nombre del contenedor en el que se encuentra el objeto que desea descargar.
3. Si el objeto que desea descargar se encuentra en una carpeta, continúe eligiendo los nombres de las carpetas hasta que vea el objeto.
4. Elija el nombre del objeto.
5. En la página de detalles de Object (Objeto), elija Download (Descargar).

Para descargar un objeto (AWS CLI)

- En AWS CLI, utilice el `get-object` comando:

```
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
```

En el siguiente ejemplo, se muestra el valor de retorno:

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

Para descargar parte de un objeto (AWS CLI)

- En AWS CLI, utilice el `get-object` comando y especifique un rango.

```
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
```

En el siguiente ejemplo, se muestra el valor de retorno:

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

Eliminación de objetos

AWS Elemental MediaStore ofrece diferentes opciones para eliminar objetos de los contenedores:

- Eliminar un objeto individual. No se aplican cargos.
- Vaciar un contenedor para eliminar todos los objetos dentro de un contenedor a la vez. Como este proceso utiliza llamadas a la API, se aplican los cargos normales de la API.
- Agregar una política de ciclo de vida de objetos para eliminar objetos cuando tengan una cierta antigüedad. No se aplican cargos.

Eliminación de un objeto

Puede eliminar objetos de manera individual mediante la consola o la AWS CLI. Como alternativa, puede agregar una política de ciclo de vida de objetos para eliminar automáticamente objetos cuando tengan una determinada antigüedad en un contenedor, o bien puede vaciar un contenedor para eliminar todos los objetos dentro de ese contenedor.

> **Note**
>
> Al eliminar el único objeto de una carpeta, AWS Elemental elimina MediaStore automáticamente la carpeta y cualquier carpeta vacía situada encima de esa carpeta. Por ejemplo, supongamos que tiene una carpeta denominada `premium` que no contiene ningún archivo, pero que contiene la subcarpeta `canada`. La subcarpeta `canada` contiene un archivo denominado `mlaw.ts`. Si elimina el archivo `mlaw.ts`, el servicio elimina las carpetas `premium` y `canada`.

Para eliminar un objeto (consola)

1. Abra la MediaStore consola en. https://console.aws.amazon.com/mediastore/
2. En la página Containers (Contenedores), elija el nombre del contenedor que tiene el objeto que desea eliminar.
3. Si el objeto que desea eliminar se encuentra en una carpeta, continúe eligiendo los nombres de las carpetas hasta que vea el objeto.
4. Elija la opción a la izquierda del nombre del objeto.
5. Elija Eliminar.

Para eliminar un objeto (AWS CLI)

- En AWS CLI, utilice el `delete-object` comando.

  Ejemplo:

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

  Este comando no tiene ningún valor de retorno.

Vaciar un contenedor

Puede vaciar un contenedor para eliminar todos los objetos almacenados en el contenedor. Como alternativa, puede agregar una política de ciclo de vida de objetos para eliminar automáticamente los

objetos una vez que tengan una determinada antigüedad en un contenedor, o bien puede eliminar los objetos individualmente.

Para vaciar un contenedor (consola)

1. Abra la MediaStore consola en https://console.aws.amazon.com/mediastore/.
2. En la página Containers (Contenedores) elija la opción del contenedor que desea vaciar.
3. Elija Empty container (Vaciar contenedor). Aparece un mensaje de confirmación.
4. Confirme que desea vaciar el contenedor introduciendo primero el nombre del contenedor en el campo de texto y eligiendo después Vaciar.

Seguridad en AWS Elemental MediaStore

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de los centros de datos y las arquitecturas de red diseñados para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre AWS usted y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la Nube de AWS. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWS programas de de . Para obtener más información sobre los programas de conformidad que se aplican a AWS Elemental MediaStore, consulte AWS Servicios dentro del alcance por programa de conformidad AWS Servicios incluidos .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También eres responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza MediaStore. Los siguientes temas muestran cómo configurarlo MediaStore para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus MediaStore recursos.

Temas

Protección de datos en AWS Elemental MediaStore

El modelo de se aplica a protección de datos en AWS Elemental MediaStore. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los Nube de AWS. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulta las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulta la publicación de blog sobre el Modelo de responsabilidad compartida de AWS y GDPR en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Se utiliza SSL/TLS para comunicarse con AWS los recursos. Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte Cómo trabajar con CloudTrail senderos en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-3 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulta Estándar de procesamiento de la información federal (FIPS) 140-3.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con MediaStore o Servicios de AWS utiliza la consola, la API o. AWS CLI AWS SDKs Cualquier dato que ingrese en etiquetas o campos de

texto de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Cifrado de datos

MediaStore cifra contenedores y objetos en reposo mediante el algoritmo AES-256 estándar del sector. Le recomendamos que lo utilice MediaStore para proteger sus datos de las siguientes maneras:

- Cree una política de contenedor para controlar los derechos de acceso a todas las carpetas y objetos de dicho contenedor. Para obtener más información, consulte the section called “Políticas de contenedor”.
- Cree una política de intercambio de recursos de origen cruzado (CORS) para permitir el acceso de origen cruzado a sus recursos de forma selectiva. MediaStore En CORS, puede permitir a aplicaciones web clientes cargadas en un dominio interactuar con los recursos de un dominio distinto. Para obtener más información, consulte the section called “Políticas CORS”.

Identity and Access Management para AWS Elemental MediaStore

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los AWS recursos. Los administradores de IAM controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar los recursos. MediaStore La IAM es una Servicio de AWS opción que puede utilizar sin coste adicional.

Temas

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo en el que se realice. MediaStore

Usuario del servicio: si utiliza el MediaStore servicio para realizar su trabajo, el administrador le proporcionará las credenciales y los permisos que necesita. A medida que vaya utilizando más MediaStore funciones para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarle a solicitar los permisos correctos al administrador. Si no puede acceder a una característica en MediaStore, consulte Solución de problemas de MediaStore identidad y acceso a AWS Elemental.

Administrador de servicios: si estás a cargo de MediaStore los recursos de tu empresa, probablemente tengas acceso total a ellos MediaStore. Su trabajo consiste en determinar a qué MediaStore funciones y recursos deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su gestionador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información sobre cómo su empresa puede utilizar la IAM MediaStore, consulteCómo MediaStore funciona AWS Elemental con IAM.

Administrador de IAM: si es un administrador de IAM, es posible que quiera conocer más detalles sobre cómo escribir políticas para administrar el acceso a MediaStore. Para ver ejemplos de políticas MediaStore basadas en la identidad que puede utilizar en IAM, consulte. Ejemplos de políticas basadas en identidad para AWS Elemental MediaStore

Autenticación con identidades

La autenticación es la forma de iniciar sesión AWS con sus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (Centro de identidades de IAM), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, su gestionador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, estás asumiendo un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte Cómo iniciar sesión Cuenta de AWS en su Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre la firma de solicitudes, consulte AWS Signature Versión 4 para solicitudes API en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactor en la Guía del usuario de AWS IAM Identity Center y Autenticación multifactor AWS en IAM en la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utiliza el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulta Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utiliza AWS IAM Identity Center. Puede crear usuarios y grupos en el Centro de identidades de IAM, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus Cuentas de AWS aplicaciones. Para obtener más información, consulta ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un usuario de IAM es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. No puedes iniciar sesión como grupo. Puedes usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos para grandes conjuntos de usuarios. Por ejemplo, puede asignar un nombre a un grupo IAMAdminsy concederle permisos para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales de larga duración permanentes; no obstante, los roles proporcionan credenciales temporales. Para obtener más información, consulte Casos de uso para usuarios de IAM en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una persona determinada. Para asumir temporalmente un rol de IAM en el AWS Management Console, puede cambiar de un rol de usuario a uno de IAM (consola). Puedes asumir un rol llamando a una operación de AWS API AWS CLI o usando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulta Métodos para asumir un rol en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles de federación, consulte Crear un rol para un proveedor de identidad de terceros (federación) en la Guía de usuario de IAM. Si utiliza el IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulta Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center .
- Permisos de usuario de IAM temporales: un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- Acceso entre cuentas: puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunas Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulta Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realizas una llamada en un servicio, es habitual que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.
  - Sesiones de acceso directo (FAS): cuando utilizas un usuario o un rol de IAM para realizar acciones en AWS ellas, se te considera principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.
  - Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio

desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

- Función vinculada al servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

- Aplicaciones que se ejecutan en Amazon EC2: puedes usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una EC2 instancia y realizan AWS CLI solicitudes a la AWS API. Esto es preferible a almacenar las claves de acceso en la EC2 instancia. Para asignar un AWS rol a una EC2 instancia y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite que los programas que se ejecutan en la EC2 instancia obtengan credenciales temporales. Para obtener más información, consulte Usar un rol de IAM para conceder permisos a las aplicaciones que se ejecutan en EC2 instancias de Amazon en la Guía del usuario de IAM.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulte Información general de políticas JSON en la Guía del usuario de IAM.

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utiliza para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción

`iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de la API AWS Management Console AWS CLI, la o la AWS API.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS empresa. Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para obtener más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso () ACLs

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios compatibles. AWS WAF ACLs Para obtener más información ACLs, consulte la descripción general de la lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). Puedes establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicios (SCPs): SCPs son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations AWS Organizations es un servicio para agrupar y administrar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilitas todas las funciones de una organización, puedes aplicar políticas de control de servicios (SCPs) a una o a todas tus cuentas. El SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una Usuario raíz de la cuenta de AWS. Para obtener más información sobre Organizations SCPs, consulte las políticas de control de servicios en la Guía del AWS Organizations usuario.
- Políticas de control de recursos (RCPs): RCPs son políticas de JSON que puedes usar para establecer los permisos máximos disponibles para los recursos de tus cuentas sin actualizar las políticas de IAM asociadas a cada recurso que poseas. El RCP limita los permisos de los recursos en las cuentas de los miembros y puede afectar a los permisos efectivos de las identidades, incluidos los permisos Usuario raíz de la cuenta de AWS, independientemente de si pertenecen a su organización. Para obtener más información sobre Organizations e RCPs incluir una lista de Servicios de AWS ese apoyo RCPs, consulte Políticas de control de recursos (RCPs) en la Guía del AWS Organizations usuario.
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades

del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del usuario de IAM.

Cómo MediaStore funciona AWS Elemental con IAM

Antes de usar IAM para administrar el acceso MediaStore, conozca las funciones de IAM disponibles para su uso. MediaStore

Características de IAM que puede usar con AWS Elemental MediaStore

Característica de IAM	MediaStore soporte
Políticas basadas en identidades	Sí
Políticas basadas en recursos	Sí
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política (específicas del servicio)	Sí
ACLs	No
ABAC (etiquetas en políticas)	Parcial
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	Sí

Característica de IAM	MediaStore soporte
Roles vinculados al servicio	No

Para obtener una visión general de cómo MediaStore funcionan otros AWS servicios con la mayoría de las funciones de IAM, consulte AWS los servicios que funcionan con IAM en la Guía del usuario de IAM.

Políticas basadas en la identidad para MediaStore

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está asociada. Para obtener más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en la identidad para MediaStore

Para ver ejemplos de políticas MediaStore basadas en la identidad, consulte. Ejemplos de políticas basadas en identidad para AWS Elemental MediaStore

Políticas basadas en recursos incluidas MediaStore

Compatibilidad con las políticas basadas en recursos: sí

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad

principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política basada en recursos concede acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para obtener más información, consulte Cross account resource access in IAM en la Guía del usuario de IAM.

> **Note**
>
> MediaStore también admite políticas de contenedores que definen qué entidades principales (cuentas, usuarios, roles y usuarios federados) pueden realizar acciones en el contenedor. Para obtener más información, consulte Políticas de contenedor.

Acciones políticas para MediaStore

Compatibilidad con las acciones de políticas: sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de MediaStore acciones, consulte Acciones definidas por AWS Elemental MediaStore en la Referencia de autorización de servicios.

Las acciones políticas MediaStore utilizan el siguiente prefijo antes de la acción:

```
mediastore
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

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

Para ver ejemplos de políticas MediaStore basadas en la identidad, consulte. Ejemplos de políticas basadas en identidad para AWS Elemental MediaStore

Recursos de políticas para MediaStore

Compatibilidad con los recursos de políticas: sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puedes hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utiliza un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

Para ver una lista de los tipos de MediaStore recursos y sus respectivos tipos ARNs, consulte Recursos definidos por AWS Elemental MediaStore en la Referencia de autorización de servicios. Para saber con qué acciones puede especificar el ARN de cada recurso, consulte Acciones definidas por AWS Elemental. MediaStore

El recurso MediaStore contenedor tiene el siguiente ARN:

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

Para obtener más información sobre el formato de ARNs, consulte Amazon Resource Names (ARNs) y AWS Service Namespaces.

Por ejemplo, para especificar el contenedor `AwardsShow` en su instrucción, utilice el siguiente ARN:

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

Claves de condición de la política para MediaStore

Compatibilidad con claves de condición de políticas específicas del servicio: sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puedes crear expresiones condicionales que utilizan operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puedes utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puedes conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulta Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Para ver una lista de claves de MediaStore condición, consulte Claves de condición de AWS Elemental MediaStore en la Referencia de autorización de servicios. Para saber con qué acciones y recursos puede usar una clave de condición, consulte Acciones definidas por AWS Elemental MediaStore.

Para ver ejemplos de políticas MediaStore basadas en la identidad, consulte. Ejemplos de políticas basadas en identidad para AWS Elemental MediaStore

ACLs in MediaStore

Soporta ACLs: No

Las listas de control de acceso (ACLs) controlan qué directores (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

ABAC con MediaStore

Compatibilidad con ABAC (etiquetas en las políticas): parcial

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a las entidades de IAM (usuarios o roles) y a muchos AWS recursos. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte Definición de permisos con la autorización de ABAC en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Utilizar credenciales temporales con MediaStore

Compatibilidad con credenciales temporales: sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluidas las que Servicios de AWS funcionan con credenciales temporales, consulta Cómo Servicios de AWS funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para obtener más información sobre el cambio de roles, consulte Cambio de un usuario a un rol de IAM (consola) en la Guía del usuario de IAM.

Puedes crear credenciales temporales manualmente mediante la AWS CLI API o. AWS A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para obtener más información, consulte Credenciales de seguridad temporales en IAM.

Permisos principales entre servicios para MediaStore

Admite sesiones de acceso directo (FAS): sí

Cuando utilizas un usuario o un rol de IAM para realizar acciones en él AWS, se te considera principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama y los que solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

Roles de servicio para MediaStore

Compatibilidad con roles de servicio: sí

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener

más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

> **Warning**
>
> Si se cambian los permisos de un rol de servicio, es posible que se interrumpa MediaStore la funcionalidad. Edite las funciones de servicio solo cuando se MediaStore proporcionen instrucciones para hacerlo.

Funciones vinculadas al servicio para MediaStore

Compatibilidad con roles vinculados al servicio: no

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puedes ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulta Servicios de AWS que funcionan con IAM. Busque un servicio en la tabla que incluya `Yes` en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Ejemplos de políticas basadas en identidad para AWS Elemental MediaStore

De forma predeterminada, los usuarios y roles no tienen permiso para crear, ver ni modificar recursos de MediaStore. Tampoco pueden realizar tareas mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS la API. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puedes añadir las políticas de IAM a roles y los usuarios puedes asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM (consola) en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos por MediaStore, incluido el ARNs formato de cada uno de los tipos de recursos, consulte Acciones, recursos y claves de condición de AWS Elemental MediaStore en la Referencia de autorización de servicios.

Temas

- Prácticas recomendadas sobre las políticas
- Mediante la consola de MediaStore
- Cómo permitir a los usuarios consultar sus propios permisos

Prácticas recomendadas sobre las políticas

Las políticas basadas en la identidad determinan si alguien puede crear MediaStore recursos de su cuenta, acceder a ellos o eliminarlos. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulta las políticas administradas por AWS o las políticas administradas por AWS para funciones de tarea en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulta Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulta Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.

- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Validación de políticas con el Analizador de acceso de IAM en la Guía del usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte Acceso seguro a la API con MFA en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Mediante la consola de MediaStore

Para acceder a la MediaStore consola de AWS Elemental, debes tener un conjunto mínimo de permisos. Estos permisos deben permitirte enumerar y ver detalles sobre los MediaStore recursos de tu cuenta Cuenta de AWS. Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No es necesario que concedas permisos mínimos de consola a los usuarios que solo realicen llamadas a la API AWS CLI o a la AWS API. En su lugar, permite el acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Para garantizar que los usuarios y los roles puedan seguir utilizando la MediaStore consola, adjunte también la política `ReadOnly` AWS gestionada MediaStore `ConsoleAccess` o la política gestionada a las entidades. Para obtener más información, consulte Adición de permisos a un usuario en la Guía del usuario de IAM:

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas gestionadas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API AWS CLI o AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Solución de problemas de MediaStore identidad y acceso a AWS Elemental

Utilice la siguiente información como ayuda para diagnosticar y solucionar los problemas más comunes que pueden surgir al trabajar con un MediaStore IAM.

Temas

- No estoy autorizado a realizar ninguna acción en MediaStore

- No estoy autorizado a realizar tareas como: PassRole
- Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis MediaStore recursos

No estoy autorizado a realizar ninguna acción en MediaStore

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `mediastore:`*`GetWidget`*.

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

En este caso, la política del usuario `mateojackson` debe actualizarse para permitir el acceso al recurso *`my-example-widget`* mediante la acción `mediastore:`*`GetWidget`*.

Si necesita ayuda, póngase en contacto con su AWS administrador. El gestionador es la persona que le proporcionó las credenciales de inicio de sesión.

No estoy autorizado a realizar tareas como: PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, las políticas deben actualizarse a fin de permitirle pasar un rol a MediaStore.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en MediaStore. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

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

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su administrador. AWS El gestionador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis MediaStore recursos

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que respaldan las políticas basadas en recursos o las listas de control de acceso (ACLs), puedes usar esas políticas para permitir que las personas accedan a tus recursos.

Para obtener más información, consulte lo siguiente:

- Para saber si MediaStore es compatible con estas funciones, consulte. Cómo MediaStore funciona AWS Elemental con IAM
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro usuario de su propiedad Cuenta de AWS en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulta Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para conocer sobre la diferencia entre las políticas basadas en roles y en recursos para el acceso entre cuentas, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Inicio de sesión y supervisión AWS Elemental MediaStore

En esta sección, se proporciona información general acerca de las opciones para registrar y monitorizar en AWS Elemental MediaStore por motivos de seguridad. Para obtener más información acerca del registro y la monitorización en MediaStore consulte Supervisión y etiquetado en AWS Elemental MediaStore.

La supervisión es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de AWS Elemental MediaStore sus AWS soluciones. Debe recopilar los datos de

supervisión de todas las partes de la AWS solución para poder depurar con mayor facilidad una falla multipunto en caso de que se produzca. AWS proporciona varias herramientas para supervisar sus MediaStore recursos y responder a posibles incidentes.

CloudWatch Alarmas Amazon

Al usar CloudWatch las alarmas, puede observar una única métrica durante un período de tiempo que especifique. Si la métrica supera un umbral determinado, se envía una notificación a un tema de Amazon SNS o a una política de Auto Scaling de AWS. CloudWatch las alarmas no invocan acciones porque se encuentran en un estado determinado. En su lugar, el estado debe haber cambiado y debe mantenerse durante el número de periodos especificado. Para obtener más información, consulte Monitorización con CloudWatch.

AWS CloudTrail registros

CloudTrail proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en AWS Elemental MediaStore. Con la información recopilada CloudTrail, puede determinar el destinatario de la solicitud MediaStore, la dirección IP desde la que se realizó la solicitud, quién la realizó, cuándo se realizó y detalles adicionales. Para obtener más información, consulte Registro de llamadas a la API con CloudTrail.

AWS Trusted Advisor

Trusted Advisor se basa en las mejores prácticas aprendidas al prestar servicio a cientos de miles de AWS clientes. Trusted Advisor inspecciona su entorno de AWS y, a continuación, hace recomendaciones cuando existen oportunidades para ahorrar dinero, mejorar la disponibilidad y el rendimiento del sistema o ayudar a cerrar las brechas de seguridad. Todos los AWS clientes tienen acceso a cinco cheques de Trusted Advisor. Los clientes con un plan de soporte empresarial o empresarial pueden ver todos los Trusted Advisor cheques.

Para obtener más información, consulte AWS Trusted Advisor.

Validación de conformidad para AWS Elemental MediaStore

Para saber si uno Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte Servicios de AWS Alcance por programa de cumplimiento Servicios de AWS de cumplimiento y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de AWS cumplimiento > Programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Cumplimiento de seguridad y gobernanza: en estas guías se explican las consideraciones de arquitectura y se proporcionan pasos para implementar las características de seguridad y cumplimiento.
- Referencia de servicios válidos de HIPAA: muestra una lista con los servicios válidos de HIPAA. No todos Servicios de AWS cumplen con los requisitos de la HIPAA.
- AWS Recursos de de cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su industria y ubicación.
- AWS Guías de cumplimiento para clientes: comprenda el modelo de responsabilidad compartida desde el punto de vista del cumplimiento. Las guías resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y orientan los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad del Sector de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).
- Evaluación de los recursos con reglas en la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub— Esto Servicio de AWS proporciona una visión completa del estado de su seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulte la Referencia de controles de Security Hub.
- Amazon GuardDuty: Servicio de AWS detecta posibles amenazas para sus cargas de trabajo Cuentas de AWS, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas y maliciosas. GuardDuty puede ayudarlo a cumplir con varios requisitos de conformidad, como el PCI DSS, al cumplir con los requisitos de detección de intrusiones exigidos por ciertos marcos de cumplimiento.
- AWS Audit Manager— Esto le Servicio de AWS ayuda a auditar continuamente su AWS uso para simplificar la gestión del riesgo y el cumplimiento de las normativas y los estándares del sector.

Resiliencia en AWS Elemental MediaStore

La infraestructura AWS global se basa en distintas zonas Regiones de AWS de disponibilidad. Regiones de AWS proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

Para obtener más información sobre las zonas de disponibilidad Regiones de AWS y las zonas de disponibilidad, consulte Infraestructura global.AWS

Además de la infraestructura AWS global, MediaStore ofrece varias funciones para ayudarlo a satisfacer sus necesidades de respaldo y resiliencia de datos.

Seguridad de la infraestructura en AWS Elemental MediaStore

Como servicio gestionado, AWS Elemental MediaStore está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte Seguridad AWS en la nube. Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte Protección de infraestructuras en un marco de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las llamadas a la API AWS publicadas para acceder a MediaStore través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puedes utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Prevención de la sustitución confusa entre servicios

El problema de la sustitución confusa es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación de identidad entre servicios puede provocar el confuso problema de un diputado. La suplantación entre servicios puedes producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puedes manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que lo ayudan a proteger sus datos para todos los servicios con entidades principales de servicio a las que se les ha dado acceso a los recursos de su cuenta.

Recomendamos usar las claves de contexto de condición `aws:SourceAccount`global `aws:SourceArn`y las claves de contexto en las políticas de recursos para limitar los permisos que AWS Elemental MediaStore otorga a otro servicio al recurso. Utiliza `aws:SourceArn` si desea que solo se asocie un recurso al acceso entre servicios. Utiliza `aws:SourceAccount` si quiere permitir que cualquier recurso de esa cuenta se asocie al uso entre servicios.

La forma más eficaz de protegerse contra el problema de la sustitución confusa es utilizar la clave de contexto de condición global de `aws:SourceArn` con el ARN completo del recurso. Si no conoce el ARN completo del recurso o si está especificando varios recursos, utilice la clave de condición de contexto global `aws:SourceArn` con caracteres comodines (*) para las partes desconocidas del ARN. Por ejemplo, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Si el valor de `aws:SourceArn` no contiene el ID de cuenta, como un ARN de bucket de Amazon S3, debe utilizar ambas claves de contexto de condición global para limitar los permisos.

El valor de `aws:SourceArn` debe ser la configuración para la que se MediaStore publican CloudWatch los registros en su región y cuenta.

El siguiente ejemplo muestra cómo puede utilizar las claves de contexto de condición `aws:SourceAccount` global `aws:SourceArn` y las claves contextuales MediaStore para evitar el confuso problema de los diputados.

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

Supervisión y etiquetado en AWS Elemental MediaStore

La supervisión es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de AWS Elemental MediaStore y del resto de sus AWS soluciones. AWS proporciona las siguientes herramientas de monitoreo para observar MediaStore, informar cuando algo está mal y tomar medidas automáticas cuando sea apropiado:

- AWS CloudTrailcaptura las llamadas a la API y los eventos relacionados realizados por su AWS cuenta o en su nombre y entrega los archivos de registro a un bucket de Amazon S3 que especifique. Puede identificar qué usuarios y cuentas llamaron AWS, la dirección IP de origen desde la que se realizaron las llamadas y cuándo se produjeron las llamadas. Para obtener más información, consulte la Guía del usuario de AWS CloudTrail.
- Amazon CloudWatch supervisa tus AWS recursos y las aplicaciones en las que ejecutas AWS en tiempo real. Puede recopilar métricas y realizar un seguimiento de las métricas, crear paneles personalizados y definir alarmas que le advierten o que toman medidas cuando una métrica determinada alcanza el umbral que se especifique. Por ejemplo, puedes CloudWatch hacer un seguimiento del uso de la CPU u otras métricas de tus EC2 instancias de Amazon y lanzar automáticamente nuevas instancias cuando sea necesario. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.
- Amazon CloudWatch Events ofrece un flujo de eventos del sistema que describen los cambios en AWS los recursos. Por lo general, AWS los servicios envían notificaciones de CloudWatch eventos a Events en cuestión de segundos, pero a veces pueden tardar un minuto o más. CloudWatch Events permite la computación automatizada basada en eventos, ya que puede escribir reglas que vigilen ciertos eventos y activen acciones automatizadas en otros AWS servicios cuando estos eventos ocurren. Para obtener más información, consulta la Guía del usuario de Amazon CloudWatch Events.
- Amazon CloudWatch Logs le permite supervisar, almacenar y acceder a sus archivos de registro desde EC2 instancias de Amazon y otras fuentes. CloudTrail CloudWatch Los registros pueden monitorear la información de los archivos de registro y notificarle cuando se alcanzan ciertos umbrales. También se pueden archivar los datos del registro en un almacenamiento de larga duración. Para obtener más información, consulta la Guía del usuario CloudWatch de Amazon Logs.

También puede asignar metadatos a sus MediaStore contenedores en forma de etiquetas. Cada etiqueta consta de una clave y un valor definidos. Las etiquetas pueden facilitar la administración,

la búsqueda y el filtrado de recursos. Puede usar etiquetas para organizar sus AWS recursos en la consola de AWS administración, crear informes de uso y facturación para todos sus AWS recursos y filtrar los recursos durante las actividades de automatización de la infraestructura.

Temas

Registrar llamadas a la MediaStore API de AWS Elemental con AWS CloudTrail

AWS Elemental MediaStore está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en MediaStore. CloudTrail captura un subconjunto de llamadas a la API MediaStore como eventos, incluidas las llamadas desde la MediaStore consola y las llamadas desde código a la MediaStore API. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos para MediaStore. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por CloudTrail, puedes determinar a qué solicitud se realizó MediaStore, la dirección IP desde la que se realizó la solicitud, quién la hizo, cuándo se hizo y más.

Para obtener más información CloudTrail, incluido cómo configurarla y habilitarla, consulta la Guía del AWS CloudTrail usuario.

Temas

MediaStoreInformación sobre AWS Elemental en CloudTrail

CloudTrail está habilitada en su AWS cuenta al crear la cuenta. Cuando se produce una actividad de eventos admitida en AWS Elemental MediaStore, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puede ver, buscar y descargar los eventos recientes en su AWS cuenta. Para obtener más información, consulte Visualización de eventos con el historial de CloudTrail eventos.

Para tener un registro continuo de los eventos de tu AWS cuenta, incluidos los eventos de tu cuenta MediaStore, crea una ruta. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De manera predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones de AWS . La ruta registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de Amazon S3 que especifique. Además, puede configurar otros AWS servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para obtener más información, consulte los temas siguientes:

- Introducción a la creación de registros de seguimiento
- CloudTrail Integraciones y servicios compatibles
- Configuración de las notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de CloudTrail registro de varias regiones y recibir archivos de CloudTrail registro de varias cuentas

AWS Elemental MediaStore admite el registro de las siguientes operaciones como eventos en los archivos de CloudTrail registro:

- `CreateContainer`
- `DeleteContainer`
- `DeleteContainerPolicy`
- `DeleteCorsPolicy`
- `DescribeContainer`
- `GetContainerPolicy`
- `GetCorsPolicy`
- `ListContainers`
- `PutContainerPolicy`
- `PutCorsPolicy`

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales del usuario raíz o del usuario:

- si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado
- Si la solicitud la realizó otro AWS servicio

Para obtener más información, consulte el Elemento userIdentity de CloudTrail.

Ejemplo: entradas del archivo de MediaStore registro de AWS Elemental

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una única solicitud de cualquier origen e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etcétera. Los archivos de registro de CloudTrail no son un rastro de la pila ordenada de las llamadas a la API públicas, por lo que no aparecen en ningún orden específico.

El siguiente ejemplo muestra una entrada de CloudTrail registro que demuestra la `CreateContainer` operación:

```
{
    "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"
}
```


Supervisión de AWS Elemental MediaStore con Amazon CloudWatch

Puede monitorizar AWS Elemental MediaStore utilizando CloudWatch, que recopila datos sin procesar y los procesa para convertirlos en métricas legibles. CloudWatch guarda las estadísticas durante 15 meses para que pueda acceder a la información histórica y obtener una mejor perspectiva del rendimiento de su aplicación o servicio web. También puede establecer alarmas que vigilen determinados umbrales y enviar notificaciones o realizar acciones cuando se cumplan dichos umbrales. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.

AWS proporciona las siguientes herramientas de monitoreo para observar MediaStore, informar cuando algo anda mal y tomar medidas automáticas cuando sea apropiado:

- Amazon CloudWatch Logs le permite monitorear, almacenar y acceder a sus archivos de registro desde AWS servicios como AWS Elemental MediaStore. Puede usar CloudWatch Logs para monitorear aplicaciones y sistemas mediante datos de registro. Por ejemplo, CloudWatch los registros pueden hacer un seguimiento del número de errores que se producen en los registros de las aplicaciones y enviarte una notificación siempre que la tasa de errores supere el umbral que especifiques. CloudWatch Logs utiliza sus datos de registro para la supervisión, por lo que no es necesario cambiar el código. Por ejemplo, puede supervisar los registros de las aplicaciones

para detectar términos literales específicos (como ValidationException ««) o contar el número de `PutObject` solicitudes que se realizaron durante un período de tiempo determinado. Cuando se encuentra el término que busca, CloudWatch Logs envía los datos a una CloudWatch métrica que especifique. Los datos de registro están cifrados mientras están en tránsito y cuando están en reposo.

- Amazon CloudWatch Events ofrece eventos del sistema que describen los cambios en AWS los recursos, como MediaStore los objetos. Por lo general, AWS los servicios envían notificaciones de CloudWatch eventos a Events en cuestión de segundos, pero a veces pueden tardar un minuto o más. Puedes configurar reglas para que coincidan con los eventos (por ejemplo, una `DeleteObject` solicitud) y dirigirlos a una o más funciones o transmisiones de destino. CloudWatch Los eventos se dan cuenta de los cambios operativos a medida que se producen. Además, CloudWatch Events responde a estos cambios operativos y toma las medidas correctivas necesarias, enviando mensajes en respuesta al entorno, activando funciones, realizando cambios y recopilando información de estado.

CloudWatch Registros

El registro de acceso proporciona registros detallados para las solicitudes que se realizan a objetos en un contenedor. Los registros de acceso son útiles para muchas aplicaciones, como, por ejemplo, auditorías de acceso y seguridad. También pueden ayudarle a conocer su base de clientes y a entender su MediaStore factura. CloudWatch Los registros se clasifican de la siguiente manera:

- Un flujo de registro es una secuencia de eventos de registro que comparten la misma fuente.
- Un grupo de registro es un grupo de flujos de registro que comparten la misma configuración de retención, monitoreo y control de acceso. Al habilitar el registro de acceso en un contenedor, MediaStore crea un grupo de registros con un nombre como`/aws/mediastore/MyContainerName`. Puede definir grupos de registro y especificar los flujos que deben incluirse en cada uno. No hay cuotas en el número de flujos de registro que pueden pertenecer a un grupo de registro.

De forma predeterminada, los registros se conservan de forma indefinida y no caducan nunca. Puede ajustar la política de retención para cada grupo de registros, manteniendo la retención indefinida o seleccionar un periodo de retención entre un día y 10 años.

Configuración de permisos para Amazon CloudWatch

Utilice AWS Identity and Access Management (IAM) para crear un rol que dé a AWS Elemental MediaStore acceso a Amazon CloudWatch. Debe realizar estos pasos para que se publiquen CloudWatch los registros de su cuenta. CloudWatch publica automáticamente las métricas de tu cuenta.

Para permitir el MediaStore acceso a CloudWatch

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación de la consola de IAM, elija Políticas, seguido de Crear política.
3. Elija la pestaña JSON y pegue la siguiente política:

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

   Esta política permite MediaStore crear grupos de registros y flujos de registros para cualquier contenedor de cualquier región de su AWS cuenta.

4. Elija Revisar política.

5. En la página Review policy (Revisar política), para Name (Nombre), escriba **`MediaStoreAccessLogsPolicy`** y después elija Create policy (Crear política).
6. En el panel de navegación de la consola de IAM, seleccione Roles y, a continuación, elija Crear rol.
7. Elija el tipo de rol Another AWS account (Otra cuenta de AWS).
8. En el campo ID de cuenta, introduzca su ID de AWS cuenta.
9. Elija Siguiente: permisos.
10. En el cuadro de búsqueda, escriba **`MediaStoreAccessLogsPolicy`**.
11. Seleccione la casilla de verificación situada junto a su nueva política y, a continuación, seleccione Next: Tags (Siguiente: Etiquetas).
12. Elija Next: Review (Siguiente: Revisar) para obtener una vista previa de su nuevo usuario.
13. En Nombre de rol, escriba **`MediaStoreAccessLogs`** y luego elija Crear rol.
14. En el mensaje de confirmación, seleccione el nombre del rol que acaba de crear (**`MediaStoreAccessLogs`**).
15. En la página Summary (Resumen) del rol, elija la pestaña Trust relationships (Relaciones de confianza).
16. Elija Editar relación de confianza.
17. En el documento de la política, cambie la entidad principal por el servicio MediaStore. Debería tener un aspecto similar al siguiente:

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

La política completa debe ser similar a la siguiente:

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

```
    }
  ]
}
```

18. Elija Actualizar política de confianza.

Habilitar el registro de acceso para un contenedor

De forma predeterminada, AWS Elemental MediaStore no recopila registros de acceso. Cuando habilitas el registro de acceso en un contenedor, MediaStore entrega a Amazon los registros de acceso de los objetos almacenados en ese contenedor CloudWatch. Los registros de acceso proporcionan registros detallados para solicitudes realizadas a cualquier objeto almacenado en el contenedor. Esta información puede incluir el tipo de solicitud, los recursos especificados en la solicitud y la hora y la fecha en que se procesó la solicitud.

> **Important**
>
> No se aplica ningún cargo adicional por habilitar el registro de acceso en un contenedor de MediaStore. Sin embargo, los archivos que el servicio le envía acumularán los cargos habituales de almacenamiento. (Puede eliminar los archivos de registro en cualquier momento). AWS no evalúa los cargos de transferencia de datos por la entrega de archivos de registro, pero sí incluye el cargo de la tasa normal de transferencia de datos para acceder a los archivos de registro.

Para habilitar el registro de acceso (AWS CLI)

- En el AWS CLI, usa el `start-access-logging` comando:

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

  Este comando no tiene ningún valor de retorno.

Deshabilitar el registro de acceso para un contenedor

Al deshabilitar el registro de acceso en un contenedor, AWS Elemental MediaStore deja de enviar los registros de acceso a Amazon CloudWatch. Estos registros de acceso no se guardan y no son recuperables.

Para deshabilitar el registro de acceso (AWS CLI)

- En el AWS CLI, utilice el `stop-access-logging` comando:

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

  Este comando no tiene ningún valor de retorno.

Solución de problemas de registro de acceso en AWS Elemental MediaStore

Si los registros de MediaStore acceso de AWS Elemental no aparecen en Amazon CloudWatch, consulte la siguiente tabla para ver las posibles causas y soluciones.

> **Note**
>
> Asegúrese de activar AWS CloudTrail los registros para facilitar el proceso de solución de problemas.

Síntoma	El Problema Might Be...	Pruebe esto...
No ves ningún CloudTrail evento, aunque los CloudTrail registros estén habilitados.	El rol de IAM no existe o tiene el nombre, permisos o política de confianza incorrectos.	Cree un rol con el nombre, permisos y política de confianza correctos. Consulte the section called “Configurar permisos para CloudWatch”.
Envió una solicitud de API `DescribeContainer` , pero la respuesta muestra que el parámetro `AccessLoggingEnabled` tiene un valor de `False`. Además, no puede ver ningún evento de CloudTrail para que el rol `MediaStoreAccessLogs` realice una llamada a `DescribeLogGroup` , `CreateLogGroup` , `DescribeL`	El rol de IAM no existe o tiene el nombre, permisos o política de confianza incorrectos.	Cree un rol con el nombre, permisos y política de confianza correctos. Consulte the section called “Configurar permisos para CloudWatch”.
	El registro de acceso no está habilitado en el contenedor.	Habilite los registros de acceso para el contenedor. Consulte the section called

Síntoma	El Problema Might Be...	Pruebe esto...
`ogStream` o `CreateLogStream` satisfactoria.		“Habilitar el registro de acceso”.

En la CloudTrail consola, aparece un evento con un error de acceso denegado relacionado con el `MediaStoreAccessLogs` rol. El CloudTrail evento puede incluir líneas como las siguientes:
 	
`"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:",`
	El rol de IAM no tiene los permisos correctos para AWS Elemental MediaStore.	Actualice el rol de IAM para que tenga los permisos y la política de confianza correctos . Consulte the section called “Configurar permisos para CloudWatch”.

Síntoma	El Problema Might Be...	Pruebe esto...
No puede ver ningún registro de un contenedor completo o de contenedores.	Es posible que su cuenta haya superado la CloudWatch cuota de grupos de registros por cuenta y región. Consulta las cuotas de los grupos de CloudWatch registros en la Guía del usuario de Amazon Logs.	En la CloudWatch consola, determine si su cuenta ha alcanzado la CloudWatch cuota de grupos de registros. Si es necesario, solicite un aumento de cuota.
Verás algunos inicios de sesión CloudWatch, pero no todos los registros que esperabas ver.	Es posible que tu cuenta haya superado la CloudWatch cuota de transacciones por segundo por cuenta y región. Consulta las cuotas `PutLogEvents` en la Guía del usuario de Amazon CloudWatch Logs.	Solicita un aumento de la cuota de CloudWatch transacciones por segundo por cuenta y región.

Formato de registro de acceso

Los archivos de registro de acceso constan de una secuencia de entradas de registro con formato JSON y cada entrada de registro representa una solicitud. El orden de los campos en el registro puede variar. A continuación se muestra un ejemplo de un archivo de registro que se compone de dos registros:

```
{
  "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
}
```


En la siguiente lista se describen los campos de entrada de registro:

AWSAccountID

El AWS identificador de la cuenta que se utilizó para realizar la solicitud.

BytesReceived

Número de bytes en el cuerpo de la solicitud que recibe el servidor de MediaStore.

BytesSent

El número de bytes en el cuerpo de la respuesta que envía el servidor de MediaStore. Este valor a menudo es el mismo que el valor del encabezado `Content-Length` incluido con las respuestas del servidor.

ContainerName

El nombre del contenedor que recibió la solicitud.

ErrorCode

El código MediaStore de error (por ejemplo,`InternalServerError`). Si no se produce ningún error, aparece el carácter `-`. Podría aparecer un código de error incluso si el código de estado es 200 (que indica una conexión cerrada o un error después de que el servidor iniciara la transmisión de la respuesta).

ExpiresAt

Fecha y hora de caducidad del objeto. Este valor se basa en la edad de caducidad establecida por una política de ciclo de vida transient data rule que se aplica al contenedor. El valor es una fecha y hora `ISO-8601` y se basa en el reloj del sistema del host que atiende la solicitud. Si la política de ciclo de vida no incorpora ninguna regla de datos transitorios que se aplique al objeto o si no hay ninguna política de ciclo de vida aplicada al contenedor, el valor de este campo es `null`. Este campo solo se aplica a las siguientes operaciones:`PutObject`, `GetObject`, `DescribeObject` y `DeleteObject`.

HTTPStatus

El código de estado HTTP numérico de la respuesta.

Operación

La operación realizada, como, por ejemplo, `PutObject` o `ListItems`.

Ruta

La ruta dentro del contenedor en el que se almacena el objeto. Si la operación no toma un parámetro de ruta, aparece el carácter `-`.

ReceivedTime

La hora del día en la que se recibe la solicitud. El valor es una fecha y hora `ISO-8601` y se basa en el reloj del sistema del host que atiende la solicitud.

Solicitante

El nombre de recurso de Amazon (ARN) de usuario de la cuenta utilizada para realizar la solicitud. Para las solicitudes sin autenticar, este valor es `anonymous`. Si la solicitud falla antes de que se complete la autenticación, es posible que este campo no figure en el registro. En ese tipo de solicitudes, `ErrorCode` podría identificar el problema de autorización.

RequestID

Cadena que AWS Elemental genera MediaStore para identificar de forma exclusiva cada solicitud.

Origen

La dirección de Internet aparente del solicitante o del principal del servicio de AWS que realiza la llamada. Si los servidores proxy y firewalls intermedios ocultan la dirección de la máquina que realiza la solicitud, el valor se establece en null.

TotalTime

La cantidad de milisegundos (ms) que la solicitud estuvo en tránsito desde la perspectiva del servidor. Este valor se mide comenzando por el momento en que se recibe su solicitud en el servicio y terminando en el momento en que se envía el último byte de la respuesta. Este valor se mide desde la perspectiva del servidor, ya que las medidas realizadas desde la perspectiva del cliente se ven afectadas por la latencia de la red.

TurnAroundTime

La cantidad de milisegundos que se MediaStore tardó en procesar la solicitud. Este valor se mide desde el momento en que se recibió el último byte de su solicitud hasta el momento en que se envió el primer byte de la respuesta.

El orden de los campos en el registro puede variar.

Los cambios del estado del registro se aplican con el tiempo

Los cambios del estado de registros de un contenedor se demoran un tiempo en implementarse efectivamente en el envío de archivos de registro. Por ejemplo, si habilita los registros para un contenedor A, algunas solicitudes que se realizan a la hora siguiente pueden registrarse, mientras que otras no. Si deshabilita el registro para el contenedor B, es posible que se sigan distribuyendo algunos registros durante la siguiente hora, mientras que otros no. En todos los casos, finalmente se aplica la nueva configuración sin que tenga que adoptar medidas adicionales.

Envío de archivos de registro de servidor según el mejor esfuerzo

Las entradas de registro de acceso se envían según el "mejor esfuerzo", es decir, en la medida que sea posible. En la mayoría de las solicitudes de registros para un contenedor debidamente configurado se envían archivos de registro. La mayoría de las entradas de registro se envían en el plazo de unas horas después de su registro, pero se pueden entregar con mayor frecuencia.

No se garantiza que los registros de acceso estén completos ni que lleguen de manera puntual. La entrada de registro de una solicitud determinada puede enviarse mucho después de que la solicitud se haya procesado realmente, y es probable no se envíe en absoluto. El objetivo de los registros de acceso es darle una idea de la naturaleza del tráfico al que se enfrenta su contenedor. Es poco usual perder entradas de registro de acceso, pero los registros de acceso no pretenden ser un recuento completo de todas las solicitudes.

Dada la naturaleza de mejor esfuerzo de la característica de los registros de acceso, los informes de uso disponibles en el portal de AWS (Informes de facturación y administración de costos en la AWS Management Console) podrían incluir una o varias solicitudes de acceso que no aparecen en un registro de acceso enviado.

Consideraciones sobre programación para el formato de registro de acceso

De vez en cuando, podemos ampliar el formato del registro de acceso añadiendo nuevos campos. Se debe escribir el código que analiza los registros de acceso para administrar los campos adicionales que no comprende.

CloudWatch Eventos

Amazon CloudWatch Events le permite automatizar sus AWS servicios y responder automáticamente a los eventos del sistema, como los problemas de disponibilidad de las aplicaciones o los cambios de recursos. Puede crear reglas sencillas para indicar qué eventos le resultan de interés, así como qué acciones automatizadas se van a realizar cuando un evento cumple una de las reglas.

> **Important**
>
> Por lo general, AWS los servicios envían notificaciones de CloudWatch eventos a Events en cuestión de segundos, pero a veces pueden tardar un minuto o más.

Cuando un archivo se carga en un contenedor o se retira de un contenedor, se activan dos eventos sucesivos en el CloudWatch servicio:

1. the section called “Evento de cambio de estado de objeto”
2. the section called “Evento de cambio de estado de contenedor”

Para obtener información sobre la suscripción a estos eventos, consulta Amazon CloudWatch.

Entre las acciones que se pueden activar automáticamente se incluyen las siguientes:

- Invocar una función AWS Lambda
- Invocar el comando Amazon EC2 Run
- Desviar el evento a Amazon Kinesis Data Streams
- Activar una máquina de AWS Step Functions estados
- Notificar un tema o una cola de Amazon SNS AWS SMS

Algunos ejemplos del uso de CloudWatch eventos con AWS Elemental MediaStore son los siguientes:

- Activación de una función de Lambda cada vez que se crea un contenedor.
- Notificación a un tema de Amazon SNS cuando se elimina un objeto.

Para obtener más información, consulta la Guía del usuario de Amazon CloudWatch Events.

Temas

- Evento de cambio de estado de MediaStore un objeto de AWS Elemental
- Evento de cambio de estado de MediaStore contenedor de AWS Elemental

Evento de cambio de estado de MediaStore un objeto de AWS Elemental

Este evento se publica cuando cambia el estado de un objeto (cuando el objeto se carga o se elimina).

> **Note**
>
> Los objetos que caducan debido a una regla de datos transitorios no emiten ningún CloudWatch evento cuando caducan.

Para obtener información sobre la suscripción a este evento, consulta Amazon CloudWatch.

Objeto cargado

```
{
  "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"
  }
}
```

Objeto eliminado

```
{
  "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"
  }
}
```

Evento de cambio de estado de MediaStore contenedor de AWS Elemental

Este evento se publica cuando cambia el estado de un contenedor (cuando el contenedor se añade o se elimina). Para obtener información sobre la suscripción a este evento, consulta Amazon CloudWatch.

Contenedor creado

```
{
  "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"
  }
}
```

Contenedor eliminado

```
{
  "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"
  }
}
```

Monitorización de AWS Elemental MediaStore con CloudWatch métricas de Amazon

Puede monitorizar AWS Elemental MediaStore utilizando CloudWatch, que recopila datos sin procesar y los procesa para convertirlos en métricas legibles. CloudWatchLas estadísticas se guardan durante 15 meses para que pueda acceder a la información histórica y obtener una mejor perspectiva del rendimiento de su aplicación o servicio web. También puede establecer alarmas que vigilen determinados umbrales y enviar notificaciones o realizar acciones cuando se cumplan dichos umbrales. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.

En el caso de AWS Elemental MediaStore, es posible que desee ver `BytesDownloaded` y enviarse un correo electrónico cuando esa métrica alcance un determinado umbral.

Para ver las métricas mediante la CloudWatch consola

Las métricas se agrupan en primer lugar por el espacio de nombres de servicio y, a continuación, por las diversas combinaciones de dimensiones dentro de cada espacio de nombres.

1. Inicie sesión en AWS Management Console y abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas.
3. En Todas las métricas, elija el espacio de nombres AWS/ MediaStore.
4. Elija la dimensión de la métrica para ver las métricas. Por ejemplo, elija `Request metrics by container` para ver las métricas de los diferentes tipos de solicitudes que se han enviado al contenedor.

Para ver las métricas mediante el AWS CLI

- En el símbolo del sistema, ejecute el siguiente comando:

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

MediaStore Métricas de AWS Elemental

En la siguiente tabla se enumeran las métricas a las que MediaStore envía AWS Elemental CloudWatch.

> **Note**
>
> Para ver las métricas, debes añadir una política de métricas al contenedor para permitir MediaStore el envío de métricas a Amazon CloudWatch.

Métrica	Descripción
`RequestCount`	Número total de solicitudes HTTP realizadas a un contenedor de MediaStore, separadas por el tipo de operación (`Put`, `Get`, `Delete`, `Describe`, `List`). Unidades: recuento Dimensiones válidas: • Nombre de contenedor • Nombre del grupo de objetos • Tipo de solicitud Estadísticas válidas: Sum
`4xxErrorCount`	El número de solicitudes HTTP MediaStore que se realizaron provocó un error de 4 veces. Unidades: recuento Dimensiones válidas: • Nombre de contenedor • Nombre del grupo de objetos

Métrica	Descripción
	• Tipo de solicitud Estadísticas válidas: Sum
`5xxErrorCount`	El número de solicitudes HTTP realizadas provocó un error de 5 veces. MediaStore Unidades: recuento Dimensiones válidas: • Nombre de contenedor • Nombre del grupo de objetos • Tipo de solicitud Estadísticas válidas: Sum
`BytesUploaded`	El número de bytes cargados para las solicitudes realizadas a un contenedor de MediaStore en las que la solicitud incluye un cuerpo. Unidades: bytes Dimensiones válidas: • Nombre de contenedor • Nombre del grupo de objetos Estadísticas válidas: Average (bytes por solicitud), Sum (bytes por periodo), Sample Count, Min, Max (igual que p100) y cualquier percentil entre p0,0 y p99,0.

Métrica	Descripción
`BytesDownloaded`	Número de bytes descargados para las solicitudes realizadas a un contenedor de MediaStore en las que la respuesta contiene un cuerpo. Unidades: bytes Dimensiones válidas: • Nombre de contenedor • Nombre del grupo de objetos Estadísticas válidas: Average (bytes por solicitud), Sum (bytes por periodo), Sample Count, Min, Max (igual que p100) y cualquier percentil entre p0,0 y p99,0.
`TotalTime`	La cantidad de milisegundos que la solicitud estuvo en tránsito desde la perspectiva del servidor. Este valor se mide desde el momento en que se MediaStore recibe la solicitud hasta el momento en que se envía el último byte de la respuesta. Este valor se mide desde la perspectiva del servidor, ya que las medidas realizadas desde la perspectiva del cliente se ven afectadas por la latencia de la red. Unidades: milisegundos Dimensiones válidas: • Nombre de contenedor • Nombre del grupo de objetos • Tipo de solicitud Estadísticas válidas: Average, Min (igual que P0,0), Max (igual que p100), cualquier percentil entre p0,0 y p100

Métrica	Descripción
`TurnaroundTime`	El número de milisegundos que se han MediaStore empleado en procesar la solicitud. Este valor se mide desde el momento en que MediaStore recibe el último byte de la solicitud hasta el momento en que envía el primer byte de la respuesta. Unidades: milisegundos Dimensiones válidas: • Nombre de contenedor • Nombre del grupo de objetos • Tipo de solicitud Estadísticas válidas: Average, Min (igual que P0,0), Max (igual que p100), cualquier percentil entre p0,0 y p100
`ThrottleCount`	El número de solicitudes HTTP realizadas a las MediaStore que se limitaron. Unidades: recuento Dimensiones válidas: • Nombre de contenedor • Nombre del grupo de objetos • Tipo de solicitud Estadísticas válidas: suma

Etiquetado de los recursos de AWS Elemental MediaStore

Una etiqueta es una etiqueta de atributo personalizada que usted asigna o que AWS asigna a un AWS recurso. Cada etiqueta tiene dos partes:

- Una clave de etiqueta (por ejemplo, `CostCenter`, `Environment` o `Project`). Las claves de etiqueta distinguen entre mayúsculas y minúsculas.

- Un campo opcional denominado valor de etiqueta (por ejemplo, `111122223333` o `Production`). Omitir el valor de etiqueta es lo mismo que utilizar una cadena vacía. Al igual que las claves de etiqueta, los valores de etiqueta distinguen entre mayúsculas y minúsculas.

Las etiquetas le ayudan a hacer lo siguiente:

- Identifica y organiza tus AWS recursos. Muchos servicios de AWS admiten el etiquetado, por lo que puede asignar la misma etiqueta a los recursos de diferentes servicios para indicar que los recursos están relacionados. Por ejemplo, puede asignar la misma etiqueta a un AWS Elemental MediaStore *`container`* que a una AWS Elemental MediaLive entrada.
- Realizar un seguimiento de los costos de AWS. Puede activar estas etiquetas en el Administración de facturación y costos de AWS panel de control. AWS utiliza las etiquetas para clasificar los costos y enviarle un informe mensual de asignación de costos. Para obtener más información, consulte Uso de etiquetas de asignación de costos en la Guía del usuario de AWS Billing.

En las siguientes secciones se proporciona más información sobre las etiquetas de AWS Elemental MediaStore.

Recursos compatibles en AWS Elemental MediaStore

Los siguientes recursos de AWS Elemental MediaStore admiten el etiquetado:

- *`container`*

Para obtener información acerca de cómo añadir y administrar etiquetas, consulte Administrar etiquetas.

AWS Elemental MediaStore no admite la función de control de acceso basada en etiquetas de AWS Identity and Access Management (IAM).

Convenciones de nomenclatura y uso de las etiquetas

Las siguientes convenciones básicas de nomenclatura y uso se aplican al uso de etiquetas con MediaStore los recursos de AWS Elemental:

- Cada recurso puede tener un máximo de 50 etiquetas.
- Para cada recurso, cada clave de etiqueta debe ser única y solo puede tener un valor.

- La longitud máxima de la clave de etiqueta es de 128 caracteres Unicode en UTF-8.
- La longitud máxima del valor de etiqueta es de 256 caracteres Unicode en UTF-8.
- Los caracteres permitidos son letras, números y espacios representables en UTF-8, además de los siguientes caracteres: . : + = @ _ / - (guion). EC2 Los recursos de Amazon permiten cualquier personaje.
- Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas. Como práctica recomendada, decida una estrategia de uso de mayúsculas y minúsculas en las etiquetas e implemente esa estrategia sistemáticamente en todos los tipos de recursos. Por ejemplo, decida si se va a utilizar `Costcenter`, `costcenter` o `CostCenter` y utilice la misma convención para todas las etiquetas. Procure no utilizar etiquetas similares con un tratamiento de mayúsculas y minúsculas incoherente.
- El `aws:` prefijo está prohibido en las etiquetas; su AWS uso está reservado. Las claves y valores de etiquetas que tienen este prefijo no se pueden editar. Las etiquetas que tengan este prefijo no cuentan para la cuota de etiquetas por recurso.

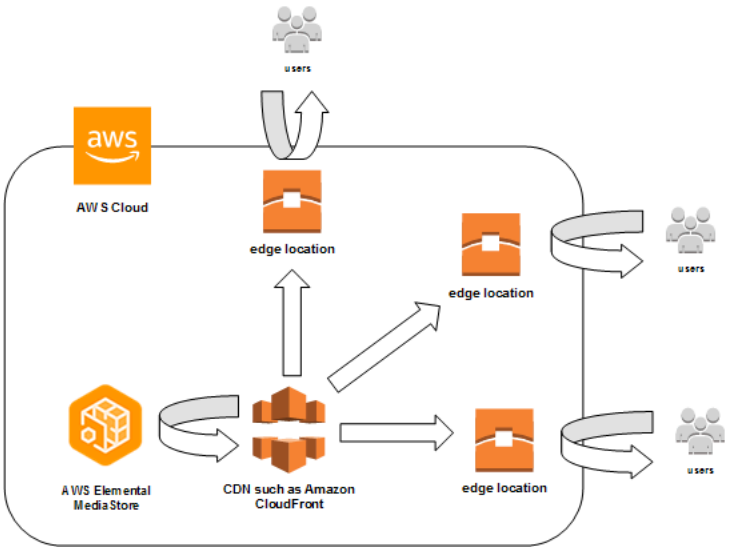
Administrar etiquetas

Las etiquetas se componen de las propiedades `Key` y `Value` de un recurso. Puede usar la API AWS CLI o la MediaStore API para añadir, editar o eliminar los valores de estas propiedades. Para obtener información sobre cómo trabajar con etiquetas, consulte las siguientes secciones de la Referencia de la MediaStore API de AWS Elemental:

- CreateContainer
- ListTagsForResource
- Recursos
- TagResource
- UntagResource

Trabajar con redes de entrega de contenido (CDNs)

Puede usar una red de entrega de contenido (CDN) como Amazon CloudFront para ofrecer el contenido que almacena en AWS Elemental MediaStore. Una CDN es un conjunto de servidores distribuidos globalmente que almacena en caché contenido, como, por ejemplo, vídeos. Cuando un usuario solicita contenido, la CDN redirige la solicitud a la ubicación de borde que ofrezca la menor latencia. Si el contenido ya se encuentra en la caché en dicha ubicación de borde, la CDN lo entrega inmediatamente. Si su contenido no se encuentra actualmente en esa ubicación perimetral, la CDN lo recupera de su origen (por ejemplo, de su MediaStore contenedor) y lo distribuye al usuario.



Temas

- Permitir que Amazon acceda CloudFront a su MediaStore contenedor de AWS Elemental
- MediaStoreLa interacción de AWS Elemental con las cachés HTTP

Permitir que Amazon acceda CloudFront a su MediaStore contenedor de AWS Elemental

Puede usar Amazon CloudFront para publicar el contenido que almacene en un contenedor de AWS Elemental MediaStore. Puede hacerlo de una de las siguientes formas:

- Uso del control de acceso al origen (OAC)- (Recomendado) Utilice esta opción si Región de AWS admite la función OAC de CloudFront.

- Uso de secretos compartidos- Utilice esta opción si no Región de AWS admite la función OAC de. CloudFront

Uso del control de acceso al origen (OAC)

Puede utilizar la función Origin Access Control (OAC) de Amazon CloudFront para proteger los MediaStore orígenes de AWS Elemental con una seguridad mejorada. Puede activar la versión 4 de AWS Signature (SiGv4) en CloudFront las solicitudes de MediaStore Origin y establecer cuándo y si CloudFront debe firmar las solicitudes. Puede acceder a la función OAC CloudFront a través de la consola APIs, el SDK o la CLI, y no hay cargos adicionales por su uso.

Para obtener más información sobre el uso de la función OAC con MediaStore, consulte Restringir el acceso a un MediaStore origen en la Guía para CloudFront desarrolladores de Amazon.

Uso de secretos compartidos

Si Región de AWS no admite la función OAC de Amazon CloudFront, puede adjuntar una política a su MediaStore contenedor de AWS Elemental que conceda acceso de lectura o superior a CloudFront.

> **Note**
>
> Le recomendamos que utilice la función OAC si la Región de AWS admite. Los siguientes procedimientos requieren que configure MediaStore y utilice CloudFront secretos compartidos para restringir el acceso a los MediaStore contenedores. Para seguir las prácticas de seguridad recomendadas, esta configuración manual requiere rotar los secretos periódicamente. Con el OAC en MediaStore los orígenes, puede indicarle que firme las solicitudes mediante SigV4 y las reenvíe a MediaStore un lugar donde se haga coincidir la firma, lo que elimina la necesidad de usar y rotar CloudFront los secretos. Esto garantiza que las solicitudes se verifiquen automáticamente antes de que se entregue el contenido multimedia, lo que hace que la entrega del contenido multimedia CloudFront sea más sencilla MediaStore y segura.

Para permitir CloudFront el acceso a su contenedor (consola)

1. Abra la MediaStore consola en https://console.aws.amazon.com/mediastore/.
2. En la página Containers (Contenedores), elija el nombre del contenedor.

Aparecerá la página de detalles del contenedor.

3. En la sección Política de contenedores, adjunta una política que conceda acceso de lectura o superior a Amazon CloudFront.

   Example

   El ejemplo de política siguiente, que es similar al ejemplo de política para Acceso público de lectura a través de HTTPS coincide con estos requisitos, ya que esto admite comandos `GetObject` y `DescribeObject` de cualquier persona que envía solicitudes a su dominio a través de HTTPS. Además, el siguiente ejemplo de política protege mejor tu flujo de trabajo, ya que solo permite el CloudFront acceso a MediaStore los objetos cuando la solicitud se produce a través de una conexión HTTPS y contiene el encabezado Referer correcto.

```
{
  "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. En la sección Container CORS policy (Política del CORS de contenedor), asigne un política que permita el nivel de acceso apropiado.

> **Note**
>
> Solo es necesaria una política del CORS si se desea proporcionar acceso a un reproductor basado en navegador.

5. Anote los detalles siguientes:

   - El punto de enlace de datos que se ha asignado a su contenedor de . Puede encontrar esta información en la sección Info (Información) de la página Containers (Contenedores). En CloudFront, el punto final de datos se denomina nombre de dominio de origen.
   - La estructura de carpetas del contenedor donde se almacenan los objetos. En CloudFront, esto se denomina ruta de origen. Tenga en cuenta que este valor es opcional. Para obtener más información sobre las rutas de origen, consulta la Guía para CloudFront desarrolladores de Amazon.

6. En CloudFront, cree una distribución que esté configurada para ofrecer contenido de AWS Elemental MediaStore. Necesitará la información que ha recopilado en el paso anterior.

Tras adjuntar la política a sus MediaStore contenedores, debe configurarla CloudFront para utilizar únicamente conexiones HTTPS para las solicitudes de origen y, además, añadir un encabezado personalizado con el valor secreto correcto.

Para configurar el acceso CloudFront a su contenedor a través de una conexión HTTPS con un valor secreto para el encabezado Referer (consola)

1. Abre la CloudFront consola.
2. En la página Origins, elige tu MediaStore origen.
3. Seleccione Editar.
4. Para el protocolo, elija solo HTTPS.
5. En la sección Añadir cabecera personalizada, seleccione Añadir encabezado.
6. Para Nombre, elija Referer. Para el valor, usa la misma `<secretValue>` cadena que usaste en tu política de contenedores.
7. Seleccione Guardar y deje que se implementen los cambios.

MediaStoreLa interacción de AWS Elemental con las cachés HTTP

AWS Elemental MediaStore almacena los objetos para que las redes de entrega de contenido (CDNs) como Amazon CloudFront puedan almacenarlos en caché de manera correcta y eficiente. Cuando un usuario final o una CDN recupera un objeto MediaStore, el servicio devuelve encabezados HTTP que afectan al comportamiento de almacenamiento en caché del objeto. (Los estándares para el comportamiento de almacenamiento en caché de HTTP 1.1 se encuentran en la sección 13). RFC2616 Estos encabezados son:

- **`ETag`** (no personalizable): el encabezado de la etiqueta de entidad es un identificador único para la respuesta que envía MediaStore . Los navegadores web CDNs y que cumplen con los estándares utilizan esta etiqueta como clave para almacenar en caché el objeto. MediaStore genera automáticamente una `ETag` para cada objeto cuando se carga. Puede ver los detalles de un objeto para determinar su ETag valor.
- **`Last-Modified`**(no personalizable): el valor de este encabezado indica la fecha y la hora en que se modificó el objeto. MediaStore genera automáticamente este valor cuando se carga el objeto.
- **`Cache-Control`** (personalizable): el valor de este encabezado controla cuánto tiempo se debe guardar en caché un objeto antes de que la CDN compruebe si se ha modificado. Puede establecer este encabezado en cualquier valor al cargar un objeto en un MediaStore contenedor mediante la CLI o la API. El conjunto completo de valores válidos se describe en la documentación HTTP/1.1. Si no estableces este valor al cargar un objeto, MediaStore no devolverá este encabezado cuando se recupere el objeto.

  Un caso de uso común para el encabezado Cache-Control es especificar una duración para almacenar en caché el objeto. Por ejemplo, supongamos que tiene un archivo de manifiesto de vídeo que un codificador sobrescribe con frecuencia. Puede establecer el `max-age` en 10 para indicar que el objeto debe almacenarlo en caché durante solo 10 segundos. O supongamos que tiene un segmento de vídeo almacenado que nunca se sobrescribirá. Puede establecer el `max-age` para este objeto en 31536000 para almacenarlo en caché durante aproximadamente 1 año.

Solicitudes condicionales

Solicitudes condicionales a MediaStore

MediaStore responde de forma idéntica a las solicitudes condicionales (utilizando encabezados de solicitud como `If-Modified-Since` y`If-None-Match`, tal como se describe en RFC7232) y a las

solicitudes incondicionales. Esto significa que cuando MediaStore recibe una `GetObject` solicitud válida, el servicio siempre devuelve el objeto, incluso si el cliente ya lo tiene.

Solicitudes condicionales a CDNs

CDNs en nombre de las que se publica contenido MediaStore pueden procesar las solicitudes condicionales devolviéndolas`304 Not Modified`, tal y como se describe en RFC7232 la sección 4.1. Esto indica que no es necesario transferir el contenido completo del objeto, porque el solicitante ya tiene un objeto que coincide con la solicitud condicional.

CDNs (y otras cachés compatibles con HTTP/1.1) basan estas decisiones en `Cache-Control` los encabezados `ETag` y encabezados que reenvían los servidores de origen. Para controlar la frecuencia con la que se CDNs consultan en los servidores de MediaStore origen las actualizaciones de los objetos recuperados repetidamente, defina `Cache-Control` los encabezados de esos objetos al cargarlos en ellos. MediaStore

Uso de este servicio con un AWS SDK

AWS Los kits de desarrollo de software (SDKs) están disponibles para muchos lenguajes de programación populares. Cada SDK proporciona una API, ejemplos de código y documentación que facilitan a los desarrolladores la creación de aplicaciones en su lenguaje preferido.

Documentación de SDK	Ejemplos de código
AWS SDK para C++	AWS SDK para C++ ejemplos de código
AWS CLI	AWS CLI ejemplos de código
AWS SDK para Go	AWS SDK para Go ejemplos de código
AWS SDK para Java	AWS SDK para Java ejemplos de código
AWS SDK para JavaScript	AWS SDK para JavaScript ejemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin ejemplos de código
AWS SDK para .NET	AWS SDK para .NET ejemplos de código
AWS SDK para PHP	AWS SDK para PHP ejemplos de código
Herramientas de AWS para PowerShell	Herramientas para ejemplos PowerShell de código
AWS SDK para Python (Boto3)	AWS SDK para Python (Boto3) ejemplos de código
AWS SDK para Ruby	AWS SDK para Ruby ejemplos de código
AWS SDK para Rust	AWS SDK para Rust ejemplos de código
AWS SDK para SAP ABAP	AWS SDK para SAP ABAP ejemplos de código
AWS SDK para Swift	AWS SDK para Swift ejemplos de código

Para obtener ejemplos específicos de este servicio, consulte Ejemplos de código para MediaStore usar AWS SDKs.

> **Ejemplo de disponibilidad**
>
> ¿No encuentra lo que necesita? Solicite un ejemplo de código a través del enlace de Enviar comentarios que se encuentra al final de esta página.

Ejemplos de código para MediaStore usar AWS SDKs

Los siguientes ejemplos de código muestran cómo usarlo MediaStore con un kit de desarrollo de AWS software (SDK).

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de código

- Ejemplos básicos de MediaStore uso AWS SDKs
  - Acciones para MediaStore usar AWS SDKs
    - Úselo CreateContainer con un AWS SDK o CLI
    - Úselo DeleteContainer con un AWS SDK o CLI
    - DeleteObjectÚselo con un AWS SDK
    - Úselo DescribeContainer con un AWS SDK o CLI
    - Úselo GetObject con un AWS SDK o CLI
    - Úselo ListContainers con un AWS SDK o CLI
    - Úselo PutObject con un AWS SDK o CLI

Ejemplos básicos de MediaStore uso AWS SDKs

Los siguientes ejemplos de código muestran cómo utilizar los conceptos básicos de AWS Elemental MediaStore with AWS SDKs.

Ejemplos

- Acciones para MediaStore usar AWS SDKs
  - Úselo CreateContainer con un AWS SDK o CLI
  - Úselo DeleteContainer con un AWS SDK o CLI
  - DeleteObjectÚselo con un AWS SDK

- Úselo DescribeContainer con un AWS SDK o CLI
- Úselo GetObject con un AWS SDK o CLI
- Úselo ListContainers con un AWS SDK o CLI
- Úselo PutObject con un AWS SDK o CLI

Acciones para MediaStore usar AWS SDKs

Los siguientes ejemplos de código muestran cómo realizar MediaStore acciones individuales con AWS SDKs. Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones para configurar y ejecutar el código.

Los siguientes ejemplos incluyen solo las acciones que se utilizan con mayor frecuencia. Para ver una lista completa, consulte la Referencia de la API de AWS Elemental MediaStore.

Ejemplos

- Úselo CreateContainer con un AWS SDK o CLI
- Úselo DeleteContainer con un AWS SDK o CLI
- DeleteObjectÚselo con un AWS SDK
- Úselo DescribeContainer con un AWS SDK o CLI
- Úselo GetObject con un AWS SDK o CLI
- Úselo ListContainers con un AWS SDK o CLI
- Úselo PutObject con un AWS SDK o CLI

Úselo **`CreateContainer`** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `CreateContainer`.

CLI

AWS CLI

Para crear un contenedor

En el siguiente ejemplo de `create-container`, se crea un nuevo contenedor vacío.

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

Salida:

```
{
    "Container": {
        "AccessLoggingEnabled": false,
        "CreationTime": 1563557265,
        "Name": "ExampleContainer",
        "Status": "CREATING",
        "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer"
    }
}
```

Para obtener más información, consulte Creación de un contenedor en la Guía del MediaStore usuario de AWS Elemental.

- Para obtener más información sobre la API, consulte CreateContainerla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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);
        }
    }
}
```

- Para obtener más información sobre la API, consulta CreateContainerla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`DeleteContainer`** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteContainer`.

CLI

AWS CLI

Para eliminar un contenedor

En el siguiente ejemplo de `delete-container`, se elimina el contenedor especificado. Un contenedor únicamente se puede eliminar si no tiene objetos.

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

Este comando no genera ninguna salida.

Para obtener más información, consulte Eliminar un contenedor en la Guía del MediaStore usuario de AWS Elemental.

- Para obtener más información sobre la API, consulte DeleteContainerla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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);
        }
    }
```

```
}
```

- Para obtener más información sobre la API, consulta DeleteContainerla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`DeleteObject`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteObject`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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();
    }
}
```

- Para obtener más información sobre la API, consulta DeleteObjectla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`DescribeContainer`** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `DescribeContainer`.

CLI

AWS CLI

Para ver los detalles de un contenedor

En el siguiente ejemplo de `describe-container`, se muestran los detalles del contenedor especificado.

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

Salida:

```
{
    "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"
    }
}
```

Para obtener más información, consulte Visualización de los detalles de un contenedor en la Guía del MediaStore usuario de AWS Elemental.

- Para obtener más información sobre la API, consulte DescribeContainerla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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 "";
    }
}
```

- Para obtener más información sobre la API, consulta DescribeContainerla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`GetObject`** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetObject`.

CLI

AWS CLI

Para descargar un objeto

En el siguiente ejemplo de `get-object`, se descarga un objeto en el punto de conexión especificado.

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

Salida:

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

Para descargar parte de un objeto

En el siguiente ejemplo de `get-object`, se descarga parte de un objeto en el punto de conexión especificado.

```
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
```

Salida:

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

Para obtener más información, consulte Descarga de un objeto en la Guía del MediaStore usuario de AWS Elemental.

- Para obtener más información sobre la API, consulte GetObjectla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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();
    }
}
```

- Para obtener más información sobre la API, consulta GetObjectla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`ListContainers`** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `ListContainers`.

CLI

AWS CLI

Para ver una lista de contenedores

En el siguiente ejemplo de `list-containers`, se muestra una lista de todos los contenedores que están asociados a la cuenta.

```
aws mediastore list-containers
```

Salida:

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

Para obtener más información, consulte Visualización de una lista de contenedores en la Guía del MediaStore usuario de AWS Elemental.

- Para obtener más información sobre la API, consulte ListContainersla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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);
        }
    }
}
```

- Para obtener más información sobre la API, consulta ListContainersla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`PutObject`** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutObject`.

CLI

AWS CLI

Para cargar un objeto

En el siguiente ejemplo de `put-object`, se carga un objeto en el contenedor especificado. Puede especificar una ruta de carpeta en la que se guardará el objeto dentro del contenedor. Si la carpeta ya existe, AWS Elemental MediaStore guarda el objeto en la carpeta. Si la carpeta no existe, el servicio la crea y, a continuación, almacena el objeto en ella.

```
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
```

Salida:

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

Para obtener más información, consulte Carga de un objeto en la Guía del MediaStore usuario de AWS Elemental.

- Para obtener más información sobre la API, consulte PutObjectla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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();
    }
}
```

- Para obtener más información sobre la API, consulta PutObjectla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Cuotas en AWS Elemental MediaStore

La consola Service Quotas proporciona información sobre las MediaStore cuotas de AWS Elemental. Además de ver las cuotas predeterminadas, puede utilizar la consola de Service Quotas para solicitar aumentos de cuota para cuotas ajustables.

En la siguiente tabla se describen las cuotas, anteriormente denominadas límites, en AWS Elemental MediaStore. Las cuotas establecen el número máximo de recursos u operaciones de servicio que puede haber en una cuenta de AWS.

> **Note**
>
> Para asignar cuotas a contenedores individuales de su cuenta, póngase en contacto con AWS Support o con el administrador de su cuenta. Esta opción puede ayudarle a dividir los límites a nivel de cuenta entre sus contenedores para evitar que un contenedor agote toda la cuota.

Recurso u operación	Cuota predeterminada	Comentarios
Contenedores	100	Número máximo de contenedores que puede crear en esta cuenta.
Niveles de carpeta	10	Número máximo de niveles de carpeta que puede crear en un contenedor. Puede crear tantas carpetas como desee, siempre que no estén anidadas más de 10 niveles dentro de un contenedor.
Carpetas	Sin límite	Puede crear tantas carpetas como desee, siempre que no estén anidadas más de 10 niveles dentro de un contenedor.
Tamaño de objeto	25 MB	El tamaño de archivo máximo de un solo objeto.

Recurso u operación	Cuota predeterminada	Comentarios
Objects	Sin límite	Puede cargar tantos objetos como desee en una carpeta o contenedor de su cuenta.
Tasa de solicitudes de API DeleteObject	100	Número máximo de solicitudes de operación que puede realizar por segundo. Las solicitudes adicionales se limitan. Puede solicitar un aumento de cuota.
Tasa de solicitudes de API DescribeObject	1 000	Número máximo de solicitudes de operación que puede realizar por segundo. Las solicitudes adicionales se limitan. Puede solicitar un aumento de cuota.
Tasa de solicitudes a la GetObjectAPI para la disponibilidad de carga estándar	1 000	Número máximo de solicitudes de operación que puede realizar por segundo. Las solicitudes adicionales se limitan. Puede solicitar un aumento de cuota.
Tasa de solicitudes a la GetObject API para determinar la disponibilidad de las subidas en streaming	25	Número máximo de solicitudes de operación que puede realizar por segundo. Las solicitudes adicionales se limitan. Puede solicitar un aumento de cuota.
Tasa de solicitudes de API ListItems	5	Número máximo de solicitudes de operación que puede realizar por segundo. Las solicitudes adicionales se limitan. Puede solicitar un aumento de cuota.

Recurso u operación	Cuota predeterminada	Comentarios
Tasa de solicitudes a la PutObject API de codificación por transferencia fragmentada (también conocida como disponibilidad de carga en streaming)	10	Número máximo de solicitudes de operación que puede realizar por segundo. Las solicitudes adicionales se limitan. Puede solicitar un aumento de cuota. En la solicitud, especifique el TPS solicitado y el tamaño de objeto medio.
Porcentaje de solicitudes de PutObjectdisponibilidad de carga estándar a la API	100	Número máximo de solicitudes de operación que puede realizar por segundo. Las solicitudes adicionales se limitan. Puede solicitar un aumento de cuota. En la solicitud, especifique el TPS solicitado y el tamaño de objeto medio.
Reglas de una política de métricas	10	Número máximo de reglas que puede incluir en una política de métrica.
Reglas en una política de ciclo de vida de objetos	10	El número máximo de reglas que puede incluir en una política de ciclo de vida de objetos.

Información MediaStore relacionada con AWS Elemental

En la siguiente tabla se enumeran los recursos relacionados que le resultarán útiles al trabajar con AWS Elemental MediaStore.

- Clases y talleres: enlaces a cursos especializados y basados en funciones, además de laboratorios personalizados que le ayudarán a perfeccionar sus AWS habilidades y adquirir experiencia práctica.
- AWS Centro para desarrolladores: explore los tutoriales, descargue herramientas y obtenga información sobre los eventos para desarrolladores. AWS
- AWS Herramientas para desarrolladores: enlaces a herramientas para desarrolladores SDKs, kits de herramientas IDE y herramientas de línea de comandos para desarrollar y administrar AWS aplicaciones.
- Centro de recursos de introducción: aprenda a configurar su aplicación Cuenta de AWS, a unirse a la AWS comunidad y a lanzar su primera aplicación.
- Tutoriales prácticos: sigue step-by-step los tutoriales para lanzar tu primera aplicación. AWS
- AWS Documentos técnicos: enlaces a una lista completa de AWS documentos técnicos, que abarcan temas como la arquitectura, la seguridad y la economía, redactados por arquitectos de AWS soluciones u otros expertos técnicos.
- AWS Support Center: el centro para crear y gestionar sus casos. AWS Support También incluye enlaces a otros recursos útiles, como foros, información técnica FAQs, estado del servicio y AWS Trusted Advisor.
- Soporte— La página web principal con información sobre Soporte un one-on-one canal de soporte de respuesta rápida que le ayudará a crear y ejecutar aplicaciones en la nube.
- Contacta con nosotros – Un punto central de contacto para las consultas relacionadas con la facturación AWS , cuentas, eventos, abuso y demás problemas.
- AWS Condiciones del sitio: información detallada sobre nuestros derechos de autor y marca comercial; su cuenta, licencia y acceso al sitio; y otros temas.

Historial de revisión de la guía del usuario

En la siguiente tabla se describe la documentación de esta versión de AWS Elemental MediaStore. Para recibir notificaciones sobre los cambios en esta documentación, puede suscribirse a una fuente RSS.

Cambio	Descripción	Fecha
Notificación del fin del soporte	Aviso de fin de soporte: el 13 de noviembre de 2025, AWS suspenderemos el soporte para AWS Elemental MediaStore. Después del 13 de noviembre de 2025, ya no podrá acceder a la MediaStore consola ni a MediaStore los recursos. Para obtener más información, visite esta publicación del blog.	12 de noviembre de 2024
Mejora del control de acceso de origen (OAC)	Se agregó información sobre cómo usar OAC con AWS Elemental MediaStore.	17 de abril de 2023
Actualizaciones de cuotas	Se han corregido la descripción y el valor de la cuota para `Rules in a Metric Policy`.	25 de octubre de 2022
ExpiresAt campo	Los registros de acceso incluyen ahora un campo `ExpiresAt` que indica la fecha y hora de caducidad del objeto en función de las reglas de datos transitorios de la política de ciclo de vida del contenedor.	16 de julio de 2020

Reglas de transición del ciclo de vida	Ahora puede agregar una regla de transición del ciclo de vida a la política de ciclo de vida de objeto que establezc a que los objetos se moverán a la clase de almacenamiento de acceso infrecuente (IA) después de que alcancen una cierta antigüedad.	20 de abril de 2020
Vaciar contenedor	Ahora puede eliminar todos los objetos dentro de un contenedor a la vez.	7 de abril de 2020
Support for Amazon CloudWatch metrics	Puedes establecer una política de métricas para determina r a qué métricas se deben MediaStore enviar CloudWatc h.	30 de marzo de 2020
Comodines en reglas de eliminación de objetos	En la política de ciclo de vida de un objeto, ahora puede utilizar un comodín en una regla de eliminación de objetos. Esto le permite especificar los archivos basados en su nombre de archivo o extensión que desea que el servicio elimine después de un determinado número de días.	20 de diciembre de 2019
Políticas de ciclo de vida de objetos	Ahora puede añadir una regla a la política de ciclo de vida de los objetos que indique un vencimiento por edad en segundos.	13 de septiembre de 2019

AWS CloudFormation apoyo	Ahora puede usar una AWS CloudFormation plantilla para crear un contenedor automáticamente. La plantilla de AWS CloudFormation administra los datos para cinco acciones de la API: creación de un contenedor, establecimiento del registro de acceso, actualización de la política de contenedor predeterminada, adición de una política de uso compartido de recursos entre orígenes (CORS) y adición de una política de ciclo de vida de objetos.	17 de mayo de 2019
Cuotas de disponibilidad de carga de streaming	Para los objetos con disponibilidad de carga de streaming (transferencia fragmentada de objetos), la operación `PutObject` no puede ser superior a 10 TPS y la operación `GetObject` no puede ser superior a 25 TPS.	8 de abril de 2019
Transferencia fragmentada de objetos	Se ha añadido soporte para transferencia fragmentada de objetos. Esta capacidad le permite especificar que un objeto está disponible para su descarga antes de que el objeto se cargue por completo.	5 de abril de 2019

Registro de acceso	AWS Elemental MediaStore ahora admite el registro de acceso, que proporciona registros detallados de las solicitudes que se realizan a los objetos de un contenedor.	25 de febrero de 2019
Políticas de ciclo de vida de objetos	Se ha agregado soporte para políticas de ciclo de vida de objetos, que rigen la fecha de vencimiento de objetos dentro del contenedor actual.	12 de diciembre de 2018
Se ha aumentado la cuota de tamaño de los objetos	La cuota de tamaño de un objeto ahora es de 25 MB.	10 de octubre de 2018
Se ha aumentado la cuota de tamaño de los objetos	La cuota de tamaño de un objeto ahora es de 20 MB.	6 de septiembre de 2018
AWS CloudTrail integration	El contenido de la CloudTrail integración se ha actualizado para adaptarlo a los cambios recientes en el CloudTrail servicio.	12 de julio de 2018
Colaboración mediante una CDN	Se agregó información sobre cómo usar AWS Elemental MediaStore con una red de entrega de contenido (CDN) como Amazon CloudFront.	14 de abril de 2018

Configuraciones CORS	AWS Elemental MediaStore ahora admite el uso compartido de recursos entre orígenes (CORS), lo que permite que las aplicaciones web cliente que se cargan en un dominio interactúen con los recursos de un dominio diferente.	7 de febrero de 2018
Nuevo servicio y guía	Esta es la versión inicial del servicio de creación y almacenamiento de vídeos, AWS Elemental MediaStore, y de la Guía del MediaStore usuario de AWS Elemental.	27 de noviembre de 2017

Note

- Los servicios AWS multimedia no están diseñados ni pensados para usarse con aplicaciones o en situaciones que requieran un rendimiento a prueba de fallos, como las operaciones de seguridad humana, los sistemas de navegación o comunicación, el control del tráfico aéreo o las máquinas de soporte vital, en las que la falta de disponibilidad, la interrupción o el fallo de los servicios podrían provocar la muerte, lesiones personales, daños a la propiedad o daños ambientales.

AWS Glosario

Para obtener la AWS terminología más reciente, consulte el AWS glosario de la Glosario de AWS Referencia.