



Guía del usuario

AWSElemental MediaStore



AWSElemental MediaStore: Guía del usuario

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es MediaStore?	1
Conceptos y terminología	1
Servicios relacionados	3
Acceder MediaStore	3
Precios	4
Regiones y puntos de enlace	4
Configuración de AWS Elemental MediaStore	6
Inscríbese en una Cuenta de AWS	6
Creación de un usuario con acceso administrativo	7
Introducción	9
Paso 1: Acceda a AWS Elemental MediaStore	9
Paso 2: Crear un contenedor	9
Paso 3: Cargar un objeto	10
Paso 4: Obtener acceso a un objeto	11
Contenedores	12
Reglas para los nombres de contenedor	12
Creación de un contenedor	12
Visualización de detalles del contenedor	14
Visualización de una lista de contenedores	15
Eliminación de un contenedor	16
Políticas	17
Políticas de contenedor	17
Visualización de una política de contenedor	18
Edición de una política de contenedor	19
Ejemplos de políticas de contenedor	20
Políticas CORS	27
Escenarios de casos de uso	27
Agregar una política de CORS	28
Visualización de una política de CORS	29
Edición de una política de CORS	30
Eliminación de una política de CORS	31
Solución de problemas	32
Ejemplos de políticas de CORS	33
Políticas de ciclo de vida de objetos	34

Componentes de una política de ciclo de vida de objetos	35
Agregar una política de ciclo de vida de objetos	42
Visualización de una política de ciclo de vida de objetos	44
Edición de una política de ciclo de vida de objetos	45
Eliminación de una política de ciclo de vida de objetos	46
Ejemplo de políticas de ciclo de vida de objetos	46
Políticas de métricas	51
Agregar una política de métricas	52
Visualización de una política de métricas	52
Edición de una política de métricas	52
Políticas de métricas de ejemplo	53
Carpetas	57
Reglas para los nombres de carpeta	58
Creación de una carpeta	58
Eliminación de una carpeta	58
Objetos	60
Carga de un objeto	60
Visualización de una lista	62
Visualización de detalles del objeto	65
Descarga de un objeto	66
Eliminación de objetos	67
Eliminación de un solo objeto	67
Vaciar un contenedor	68
Seguridad	70
Protección de datos	71
Cifrado de datos	72
Identity and Access Management	72
Público	73
Autenticación con identidades	73
Administración de acceso mediante políticas	77
Cómo MediaStore funciona AWS Elemental con IAM	80
Ejemplos de políticas basadas en identidades	87
Resolución de problemas	90
Registro y monitorización	92
CloudWatch Alarmas Amazon	92
AWS CloudTrail registros	93

AWS Trusted Advisor	93
Validación de conformidad	93
Resiliencia	94
Seguridad de infraestructuras	95
Prevencción de la sustitución confusa entre servicios	95
Monitoreo y etiquetado	98
Registrar llamadas a la API con CloudTrail	99
MediaStoreInformación en CloudTrail	99
Ejemplo: entradas de archivos de registro	101
Monitorización con CloudWatch	102
CloudWatch Registros	103
CloudWatch Eventos	113
Métricas de CloudWatch	117
Etiquetado	121
Recursos compatibles en AWS Elemental MediaStore	122
Convenciones de nomenclatura y uso de las etiquetas	122
Administración de etiquetas	123
Uso de CDN	124
Cómo permitir que CloudFront obtenga acceso a un contenedor	124
Uso del control de acceso a origen (OAC)	125
Uso de secretos compartidos	125
Interacción de MediaStore con cachés HTTP	128
Solicitudes condicionales	128
Trabajando con AWS SDKs	130
Ejemplos de código	132
Conceptos básicos	132
Acciones	133
Cuotas	156
Información relacionada	159
Historial de documentos	160
Glosario de AWS	165
.....	clxvi

¿Qué es AWS Elemental MediaStore?

AWS Elemental MediaStore es un servicio de creación y almacenamiento de vídeo que ofrece el alto rendimiento y la coherencia inmediata necesarios para la creación en directo. Con él MediaStore, puede administrar los activos de vídeo como objetos en contenedores para crear flujos de trabajo 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 excelente opción para almacenar archivos de vídeo fragmentados cuando se necesita una gran coherencia, lecturas y escrituras de baja latencia y la capacidad de gestionar grandes volúmenes de solicitudes simultáneas. Si no distribuyes vídeos en directo, considera utilizar [Amazon Simple Storage Service \(Amazon S3\)](#) en su lugar.

Temas

- [MediaStore Conceptos y terminología de AWS Elemental](#)
- [Servicios relacionados](#)
- [Acceso a AWS Elemental MediaStore](#)
- [Precios de AWS Elemental MediaStore](#)
- [Regiones y puntos de enlace de AWS Elemental MediaStore](#)

MediaStore Conceptos y terminología de AWS 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, indicado como URL raíz de 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 activo, 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 origen 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 ARN:

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

Servicios relacionados

- Amazon CloudFront es un servicio de red global de entrega de contenido (CDN) que entrega datos y vídeos de forma segura a sus espectadores. Utilice CloudFront para enviar contenido con el mejor desempeño posible. Para obtener más información, consulta la [Guía para CloudFront desarrolladores de Amazon](#).
- AWS CloudFormation es un servicio que le ayuda a modelar y configurar los recursos de AWS. Usted crea una plantilla que describe todos los AWS recursos que desea (como los MediaStore contenedores) y AWS CloudFormation se encarga de aprovisionar y configurar esos recursos por usted. No es necesario crear y configurar individualmente los recursos de AWS ni averiguar qué depende de qué. AWS CloudFormation se encarga de todo eso. Para obtener más información, consulte la [Guía del usuario de AWS CloudFormation](#).
- AWS CloudTrail es un servicio que le permite supervisar las llamadas realizadas a la CloudTrail API de su cuenta, incluidas las llamadas realizadas desde la consola de administración de AWS y otros servicios. 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. Usa CloudWatch Events para realizar un seguimiento de los cambios en el estado de los contenedores y objetos que contienen MediaStore. Para obtener más información, consulta 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 los recursos de AWS. Utilice IAM para controlar quién puede usar los recursos de AWS (autenticación), así como cuáles de ellos pueden usar 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 MediaStore

Puede acceder 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 MediaStore mediante la consola:

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

- AWS Command Line Interface— Para obtener más información, consulte la [Guía AWS Command Line Interface del usuario](#). Para acceder MediaStore mediante el punto final CLI:

```
aws mediastore
```

- MediaStore API: si utilizas un lenguaje de programación para el que no hay un SDK disponible, consulta la [referencia de la AWS Elemental MediaStore API](#) para obtener información sobre las acciones de la API y sobre cómo realizar solicitudes a la API. Para acceder MediaStore mediante el punto final de la API REST:

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

- SDK de AWS: si utiliza un lenguaje de programación para el que AWS proporciona un SDK, puede usar un SDK para obtener acceso a MediaStore. Los SDK simplifican la autenticación, se integran fácilmente con su entorno de desarrollo y proporcionan acceso sencillo a los comandos de MediaStore. 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 MediaStore

Al igual que con otros AWS productos, 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 los [MediaStore precios de AWS Elemental](#).

Regiones y puntos de enlace de AWS Elemental MediaStore

Para reducir la latencia de los datos en sus aplicaciones, MediaStore ofrece un punto de enlace regional para realizar su solicitud:

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

Para ver la lista completa de las regiones MediaStore de AWS disponibles, consulte los [MediaStore puntos de enlace y las cuotas de AWS Elemental](#) en la Referencia general de AWS.

Configuración de AWS Elemental MediaStore

Esta sección lo guía a través de los pasos necesarios para configurar los usuarios para que accedan a AWS Elemental MediaStore. Para obtener información general y adicional sobre la administración de identidades y accesos para MediaStore, consulte [Identity and Access Management para AWS Elemental MediaStore](#).

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

Temas

- [Inscríbese en una Cuenta de AWS](#)
- [Creación de un usuario con acceso administrativo](#)

Inscríbese en una Cuenta de AWS

Si no tienes un Cuenta de AWS, complete los pasos siguientes para crear uno.

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 e indicar un código de verificación en el teclado del teléfono.

Cuando te registras en una Cuenta de AWS, un Usuario raíz de la cuenta de AWS se crea. El usuario root tiene acceso a todos Servicios de AWS y los recursos de la 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 suscribirse a una Cuenta de AWS, asegure su Usuario raíz de la cuenta de AWS, habilitar AWS IAM Identity Center y cree un usuario administrativo para no utilizar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión en la [AWS Management Console](#) como propietario de la cuenta seleccionando el usuario root e introduciendo su Cuenta de AWS dirección de correo electrónico. En la siguiente página, escriba su contraseña.

Para obtener ayuda para iniciar sesión con un usuario root, consulte [Iniciar sesión como usuario root](#) en la AWS Sign-In Guía del usuario.

2. Activa la autenticación multifactorial (MFA) para tu usuario root.

Para obtener instrucciones, consulte [Habilitar un MFA dispositivo virtual para su Cuenta de AWS usuario root \(consola\)](#) en la Guía IAM del usuario.

Creación de un usuario con acceso administrativo

1. Habilite IAM Identity Center.

Para obtener instrucciones, consulte [Habilitar AWS IAM Identity Center](#) en la AWS IAM Identity Center Guía del usuario.

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

Para ver un tutorial sobre el uso de 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 AWS IAM Identity Center Guía del usuario.

Iniciar sesión como usuario con acceso de administrador

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

Para obtener ayuda para iniciar sesión con un usuario de IAM Identity Center, consulte [Iniciar sesión en el AWS acceda al portal](#) en el AWS Sign-In Guía del 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 con privilegios mínimos.

Para obtener instrucciones, consulte [Crear un conjunto de permisos en el](#) AWS IAM Identity Center Guía del usuario.

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

Para obtener instrucciones, consulte [Añadir grupos](#) en la AWS IAM Identity Center Guía del usuario.

Introducción a AWS Elemental MediaStore

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

Temas

- [Paso 1: Acceda a AWS Elemental MediaStore](#)
- [Paso 2: Crear un contenedor](#)
- [Paso 3: Cargar un objeto](#)
- [Paso 4: Obtener acceso a un objeto](#)

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 la MediaStore consolaAWS Management Console y ábrala 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

Los contenedores de AWS Elemental se utilizan MediaStore para almacenar las carpetas y los 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. Elige 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 MediaStore la 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 ElementalMediaStore

Los contenedores de MediaStore se utilizan para almacenar las carpetas y los 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 acerca de los cargos de, consulte [AWS ElementalMediaStorePrecios](#).

Temas

- [Reglas para los nombres de contenedor](#)
- [Creación de un contenedor](#)
- [Visualización de los detalles de un contenedor](#)
- [Visualización de una lista de contenedores](#)
- [Eliminación de un contenedor](#)

Reglas para los nombres de contenedor

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

- El nombre debe ser único dentro de la 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 utilizando una plantilla de AWS CloudFormation. 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 el iconoMediaStoreConsola de <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. ElegirCreación de contenedor. AWS ElementalMediaStoreañade el contenedor nuevo 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 la AWS CLI, utilice el comando create-container:

```
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 el icono MediaStore Consola de <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 la AWS CLI, utilice el comando `describe-container`:

```
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://aaabbbcccddee.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 el icono **MediaStore Consola** de <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 la AWS CLI, utilice el comando `list-containers`.

```
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 el icono MediaStore Consola de <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 (Delete).

Para eliminar un contenedor (AWS CLI)

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

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

Este comando no tiene ningún valor de retorno.

Políticas en AWS ElementalMediaStore

Puede aplicar una o varias de estas políticas a AWS ElementalMediaStoreContenedor:

- [Política de contenedor](#)- Establece los derechos de acceso a todas las carpetas y objetos dentro del contenedor. MediaStoreestablece una política predeterminada que permite a los usuarios realizar todoMediaStoreoperaciones 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 uso compartido de recursos entre orígenes \(CORS\)](#)- Permite a aplicaciones web clientes de un dominio interactuar con los recursos de un dominio distinto. MediaStoreno establece una política CORS predeterminada.
- [Política de métricas](#)- PermiteMediaStorepara enviar métricas a AmazonCloudWatch. MediaStoreno establece una política de métricas predeterminada.
- [Política de ciclo de vida de objetos](#)- Controla cuánto tiempo permanecen los objetos en unMediaStoreContenedor. MediaStoreno establece una política de ciclo de vida de objetos predeterminada.

Políticas de contenedores en AWS ElementalMediaStore

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 asocia automáticamente a todos los contenedores nuevos, permite el acceso a todos los AWS ElementalMediaStoreoperaciones 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 utilizar 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 el icono MediaStore Consola de 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 la AWS CLI, utilice el comando `get-container-policy`:

```
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 el icono MediaStoreConsola de en <https://console.aws.amazon.com/mediastore/>.
2. En la página Containers (Contenedores), elija el nombre del contenedor.
3. Elija Edit policy (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 Save (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 la AWS CLI, utilice el comando `put-container-policy`:

```
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 Valor predeterminado](#)
- [Ejemplo de política de contenedor: política Acceso de lectura público a través de HTTPS](#)
- [Ejemplo de política de contenedor: política Acceso de lectura público a través de HTTP o HTTPS](#)
- [Ejemplo de política de contenedor: política Acceso de lectura entre cuentas con HTTP habilitado](#)
- [Ejemplo de política de contenedor: política Acceso de lectura entre cuentas a través de HTTPS](#)
- [Ejemplo de política de contenedor: política Acceso de lectura entre cuentas a un rol](#)
- [Ejemplo de política de contenedor: política Acceso completo entre cuentas a un rol](#)
- [Ejemplo de política de contenedor: política Acceso restringido a direcciones IP específicas](#)

Ejemplo de política de contenedor: política Valor predeterminado

Al crear un contenedor, AWS ElementalMediaStorele asocia 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, sí puede [editar la política](#) en el contenedor si los permisos de la directiva predeterminada no se alinean con los permisos que desea utilizar 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: política 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: política 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: política 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: política Acceso de lectura entre cuentas a través de HTTPS

Este ejemplo de política permite el acceso a la `GetObject` y `DescribeObject` en cualquier objeto (tal como especifica el carácter `*` al final de la ruta de recurso) que pertenecen a la cuenta raíz del especificado `<other acct number>`. 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: política 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: política 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: política Acceso restringido a direcciones IP específicas

Este ejemplo de política permite el acceso a todos los AWS ElementalMediaStoreoperaciones 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 instrucción identifica el rango 198.51.100.* de direcciones IP permitidas en formato de Protocolo de Internet versión 4 (IPv4), 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 valores de IPv4 aws:sourceIp 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 del uso compartido de recursos entre orígenes (CORS) en AWS ElementalMediaStore

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 soporte CORS en AWS ElementalMediaStore, es posible desarrollar aplicaciones web del lado del cliente completas con MediaStore y permitir selectivamente el acceso de origen cruzado a su MediaStore de AWS.

Note

Si utiliza Amazon CloudFront para distribuir contenido de un contenedor que tiene una política CORS, asegúrese de [configurar la distribución para AWS ElementalMediaStore](#) (incluido el paso para 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 se puede habilitar el CORS con AWS ElementalMediaStore consola o mediante programación mediante la MediaStore API REST y SDK de AWS.

Temas

- [Escenarios de casos de uso de CORS](#)
- [Agregar una política de CORS a un contenedor](#)
- [Visualización de una política de CORS](#)
- [Edición de una política de CORS](#)
- [Eliminación de una política de CORS](#)
- [Solución de problemas de CORS](#)
- [Ejemplos de políticas de CORS](#)

Escenarios de casos de uso de CORS

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

- Escenario 1: Suponga que distribuye streaming de vídeo en directo en un AWS ElementalMediaStore Contenedor 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`. Si desea utilizar un JavaScriptreproductor de vídeo para acceder a los vídeos procedentes de este contenedor a través de no autenticados GET y PUT solicitudes. Normalmente, un navegador se bloquearía JavaScript de permitir esas solicitudes, pero es posible establecer una política del CORS en el contenedor para que se habiliten de manera explícita estas solicitudes de `www.example.com`.

- Escenario 2: Suponga que desea alojar la misma transmisión en directo que en el Escenario 1 desde su MediaStore contenedor, pero desea 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 AWS Elemental MediaStore Contenedor. 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 el icono MediaStore Consola de 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 la AWS CLI, utilice el comando `put-cors-policy`.

```
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 el icono `MediaStoreConsola` de 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 la AWS CLI, utilice el comando `get-cors-policy`:

```
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 el icono `MediaStoreConsola` de 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 la AWS CLI, utilice el comando `put-cors-policy`.

```
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 el iconoMediaStoreConsola de 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 la AWS CLI, utilice el comando `delete-cors-policy`:

```
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 no se encuentra el encabezado, AWS ElementalMediaStore no considera la solicitud como una solicitud entre orígenes y no devuelve los encabezados de respuesta 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

- [Ejemplos de políticas del CORS: Acceso de lectura para cualquier dominio](#)
- [Ejemplos de políticas del CORS: Acceso de lectura para un dominio específico](#)

Ejemplos de políticas del CORS: Acceso de lectura para cualquier dominio

La siguiente política permite a una página web de cualquier dominio recuperar contenido desde un `AWS ElementalMediaStoreContenedor`. 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
}
]
```

Ejemplos de políticas del CORS: Acceso de lectura para un dominio específico

La siguiente política permite una página web de `https://www.example.com` para recuperar contenido de su `AWS ElementalMediaStore` contenedor. 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 de ciclo de vida de objetos en AWS ElementalMediaStore

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 especificada, `AWS ElementalMediaStore` elimina los objetos. Puede eliminar objetos cuando ya no sean necesarios para ahorrar los costes de almacenamiento.

También puede especificar que MediaStore debe mover los objetos a la clase de almacenamiento de acceso infrecuente (IA) después de que alcancen una cierta edad. 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 [Precios de MediaStore](#).

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](#)
- [Agregar una política de ciclo de vida de objetos a un contenedor](#)
- [Visualización de una política de ciclo de vida de objetos](#)
- [Edición de una política de ciclo de vida de objetos](#)
- [Eliminación de una política de ciclo de vida de objetos](#)
- [Ejemplo de políticas de ciclo de vida de objetos](#)

Componentes de una política de ciclo de vida de objetos

Las políticas de ciclo de vida de objetos rigen el tiempo que los objetos permanecen en un AWS ElementalMediaStoreContenedor. 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. MediaStore tarda hasta 20 minutos en 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.m3u8` 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 transitoria no emiten un `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. MediaStore tarda hasta 20 minutos en aplicar la nueva política, pero los objetos pueden tardar hasta 24 horas en eliminarse del contenedor.

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 obtener 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. MediaStore tarda hasta 20 minutos en aplicar la nueva política, pero los objetos pueden tardar hasta 24 horas en eliminarse del contenedor.

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 a AWS ElementalMediaStore para eliminar objetos almacenados en el `LiveEvents/Football` una vez que tengan más de 28 días de antigüedad.
- 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 objetos en el `Football` que coincide con el patrón `index*.m3u8` después de tener 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. Puede establecer una fecha de vencimiento y después de la fecha de vencimiento: AWS ElementalMediaStore 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 el icono MediaStore Consola de: <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 la AWS CLI, utilice el comando `put-lifecycle-policy`:

```
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 el icono MediaStore Consola de: <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 la AWS CLI, utilice el comando `get-lifecycle-policy`:

```
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 el icono MediaStoreConsola de: <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 la AWS CLI, utilice el comando `put-lifecycle-policy`:

```
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 el icono MediaStore Consola de: <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 la AWS CLI, utilice el comando `delete-lifecycle-policy`:

```
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: Caducidad en cuestión de segundos](#)
- [Ejemplo de política de ciclo de vida de objeto: Caducidad en días](#)
- [Ejemplo de política de ciclo de vida de objeto: Transición a clase de almacenamiento de acceso infrecuente](#)
- [Ejemplo de política de ciclo de vida de objeto: Múltiples reglas](#)
- [Ejemplo de política de ciclo de vida de objeto: Vaciar contenedor](#)

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

La siguiente política especifica que MediaStore elimina objetos que coinciden con 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: Caducidad en días

La siguiente política especifica que MediaStore elimina objetos que coinciden con 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 MediaStore mueve objetos a la clase de almacenamiento de acceso infrecuente (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: Vaciar contenedor

La siguiente política de ciclo de vida de objeto especifica que MediaStore elimina todos los objetos del contenedor, incluidas las carpetas y subcarpetas, 1 día después de que se agregan al contenedor. Si el contenedor contiene objetos antes de aplicar esta política, MediaStore elimina los objetos 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 para permitir que AWS Elemental MediaStore envíe métricas a Amazon CloudWatch. Se necesitan hasta 20 minutos para que la nueva política surta efecto. Para obtener una descripción de cada MediaStore métrica, consulte [MediaStore 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 dictan 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 dictan 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 Save (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

Este ejemplo de política indica que AWS Elemental MediaStore debe enviar métricas a Amazon a CloudWatch 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 carpeta `baseball/saturday` tienen una CloudWatch dimensión `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

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

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

Este ejemplo de política indica que AWS Elemental MediaStore debe enviar métricas a Amazon a CloudWatch nivel de contenedor. Además, también MediaStore debe enviar las 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

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

Las métricas de MediaStore las solicitudes a las `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 ElementalMediaStore

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 ElementalMediaStore almacena el objeto en la carpeta. Si la carpeta no existe, el servicio la crea y, a continuación, almacena el objeto en ella.

Supongamos que tiene un contenedor denominado `movies`, y subes un archivo llamado `mlaw.ts` con el camino `premium/canada`. AWS ElementalMediaStore almacena el objeto en la subcarpeta `canada` debajo de 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 ElementalMediaStore elimina automáticamente una carpeta cuando se elimina el último objeto de dicha 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](#)
- [Creación de una carpeta](#)
- [Eliminación de una carpeta](#)

Reglas para los nombres de carpeta

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

- El nombre solo debe contener los siguientes caracteres: letras mayúsculas (A-Z), minúsculas (a-z), números (0-9), puntos (.), guiones (-), tildes (~), guiones (_), signo de igual (=) y dos puntos (:).
- El nombre debe tener al menos un carácter de longitud. Nombres de carpetas vacíos (tales como `folder1//folder3/`) no se permiten.
- 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 ElementalMediaStore 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 ElementalMediaStore elimina automáticamente una carpeta cuando se elimina el último objeto de dicha 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 de AWS ElementalMediaStore

AWS ElementalMediaStore los recursos de reciben el nombre objects. 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 los objetos de forma local antes de cargarlos en MediaStore.
- **Descargar:** copiar un objeto desde MediaStore a 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](#)
- [Visualización de una lista de objetos](#)
- [Visualización de los detalles de un objeto](#)
- [Descarga de un objeto](#)
- [Eliminación de objetos](#)

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 ElementalMediaStore 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 ElementalMediaStore](#).

Puede utilizar la consola o la MediaStore de AWS CLI para cargar objetos.

MediaStore admite la transferencia fragmentada de objetos, lo que reduce la latencia haciendo que un objeto esté disponible para su descarga mientras se esté 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 especifica este encabezado en su 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. Abra el icono MediaStore Consola de 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 la AWS CLI, utilice el comando `put-object`. 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 utilizar AWS ElementalMediaStoreConsola de para ver los elementos (objetos y carpetas) almacenados en el nivel superior de un contenedor o una carpeta. Los elementos almacenados en una subcarpeta del contenedor o la carpeta actual no se mostrarán. Puede utilizar la AWS CLI para ver una lista de los objetos y las carpetas que contiene un contenedor, independientemente del número de carpetas o subcarpetas que contenga el contenedor.

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

1. Abra el iconoMediaStoreConsola de 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 el icono `MediaStoreConsola` de 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 la AWS CLI, utilice el comando `list-items`:

```
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 la AWS CLI, utilice el comando `list-items` y especifique el nombre de carpeta 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 ElementalMediaStore almacena información detallada sobre él como, por ejemplo, la fecha de modificación, el tamaño del contenido, la 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 el icono MediaStore Consola de 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 la AWS CLI, utilice el comando `describe-object`:

```
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",
```


Note

Cuando se elimina el único objeto de una carpeta, AWS ElementalMediaStoreelimina automáticamente la carpeta y cualquier carpeta vacía situada por encima de dicha 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 el iconoMediaStoreConsola de 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 (Delete).

Para eliminar un objeto (AWS CLI)

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

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 el icono MediaStore Consola de 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 el nombre del contenedor en el campo de texto y, a continuación, elija Vacío.

Seguridad en AWS Elemental MediaStore

Seguridad en la nube en 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 y tú. 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 se ejecuta AWS servicios en el 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 del [AWS Programas de cumplimiento](#) . Para obtener más información sobre los programas de cumplimiento que se aplican a AWS Elemental MediaStore, consulte [AWS Servicios incluidos en el ámbito de aplicación del programa de cumplimiento](#) .
- Seguridad en la nube: su responsabilidad viene determinada por la AWS servicio que utiliza. Usted también es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos 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 usar otros AWS servicios que le ayudan a supervisar y proteger sus MediaStore recursos.

Temas

- [Protección de datos en AWS Elemental MediaStore](#)
- [Identity and Access Management para AWS Elemental MediaStore](#)
- [Inicio de sesión y supervisión AWS Elemental MediaStore](#)
- [Validación de la conformidad de AWS Elemental MediaStore](#)
- [La resiliencia en AWS Elemental MediaStore](#)
- [Seguridad de la infraestructura en Elemental AWS MediaStore](#)
- [Prevención de la sustitución confusa entre servicios](#)

Protección de datos en AWS Elemental MediaStore

La AWS [modelo de responsabilidad compartida](#) se aplica a la 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. Usted es responsable de mantener el control sobre el contenido que está alojado en esta infraestructura. También es responsable de las tareas de configuración y administración de la seguridad del Servicios de AWS que utilices. Para obtener más información sobre la privacidad de los datos, consulte la sección [Privacidad de datos FAQ](#). Para obtener información sobre la protección de datos en Europa, consulte la [AWS Modelo de responsabilidad compartida y entrada de GDPR](#) blog sobre AWS Blog de seguridad.

Para fines de protección de datos, le recomendamos que proteja Cuenta de AWS 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:

- Utilice la autenticación multifactorial (MFA) con cada cuenta.
- Utilice SSL/TLS para comunicarse con AWS recursos. Necesitamos TLS 1.2 y recomendamos TLS 1.3.
- Configure API y registre la actividad de los usuarios con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte [Trabajar con CloudTrail senderos](#) en la AWS CloudTrail Guía del usuario.
- Use AWS soluciones de cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utilice servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita entre FIPS 140 y 3 módulos criptográficos validados para acceder AWS a través de una interfaz de línea de comandos o API, utilice un FIPS punto final. Para obtener más información sobre los FIPS puntos finales disponibles, consulte la [Norma Federal de Procesamiento de Información \(FIPS\) 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 u otros MediaStore Servicios de AWS utilizando la consola API, AWS CLI, o AWS SDKs. Cualquier dato que ingrese en etiquetas o

campos de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, le recomendamos encarecidamente que no incluya información sobre las credenciales URL para validar su solicitud a ese servidor.

Cifrado de datos

MediaStore cifra los contenedores y los 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 entre orígenes (CORS) para permitir el acceso de origen cruzado a sus recursos de forma selectiva. MediaStore Con ellaCORS, puede permitir que las aplicaciones web cliente que se cargan en un dominio interactúen con los recursos de un dominio diferente. 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 un Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a AWS recursos. IAM los administradores controlan quién puede autenticarse (iniciar sesión) y quién está autorizado (tiene permisos) para usar MediaStore los recursos. IAM es un Servicio de AWS que puede utilizar sin coste adicional.

Temas

- [Público](#)
- [Autenticación con identidades](#)
- [Administración de acceso mediante políticas](#)
- [Cómo MediaStore funciona AWS Elemental con IAM](#)
- [Ejemplos de políticas basadas en la identidad para Elemental AWS MediaStore](#)
- [Solución de problemas de MediaStore identidad y acceso a AWS Elemental](#)

Público

Cómo se usa AWS Identity and Access Management (IAM) difiere 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 ayudarlo a solicitar los permisos correctos al administrador. Si no puede acceder a una función en MediaStore, consulte [Solución de problemas de MediaStore identidad y acceso a AWS Elemental](#).

Administrador de servicios: si está a cargo de MediaStore los recursos de su empresa, probablemente tenga acceso total a ellos MediaStore. Su trabajo consiste en determinar a qué MediaStore funciones y recursos deben acceder los usuarios del servicio. A continuación, debe enviar solicitudes a su IAM administrador para cambiar los permisos de los usuarios del servicio. Revise la información de esta página para comprender los conceptos básicos de IAM. Para obtener más información sobre cómo su empresa puede utilizar IAM con MediaStore, consulte [Cómo MediaStore funciona AWS Elemental con IAM](#).

IAM administrador: si es IAM administrador, puede que desee obtener más información sobre cómo redactar políticas para administrar el acceso a ellas MediaStore. Para ver ejemplos de políticas MediaStore basadas en la identidad que puede utilizar IAM, consulte. [Ejemplos de políticas basadas en la identidad para Elemental AWS MediaStore](#)

Autenticación con identidades

La autenticación es la forma de iniciar sesión en AWS utilizando tus credenciales de identidad. Debe estar autenticado (iniciar sesión en AWS) como Usuario raíz de la cuenta de AWS, como IAM usuario o asumiendo un IAM rol.

Puede iniciar sesión en AWS como identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios de (IAM Identity Center), 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, el administrador configuró previamente la federación de identidades mediante roles. IAM Cuando accedes AWS al usar la federación, está asumiendo indirectamente un rol.

Según el tipo de usuario que sea, puede iniciar sesión en AWS Management Console o el AWS portal de acceso. Para obtener más información sobre cómo iniciar sesión en AWS, consulta [Cómo iniciar sesión en tu Cuenta de AWS](#) en la AWS Sign-In Guía del usuario.

Si accedes 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 usa AWS herramientas, debe firmar las solicitudes usted mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte [Firmar AWS APIsolicitudes](#) en la Guía IAM del usuario.

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

Cuenta de AWS usuario raíz

Al crear un Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos los Servicios de AWS y los recursos de la cuenta. Esta identidad se denomina Cuenta de AWS usuario root y se accede a él iniciando sesión con la dirección de correo electrónico y la contraseña que utilizó para crear la cuenta. Recomendamos encarecidamente que no utilice 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 tareas que requieren que inicie sesión como usuario root, consulte [Tareas que requieren credenciales de usuario root](#) en la Guía del IAM usuario.

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 identidades o cualquier usuario 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 la administración centralizada del acceso, le recomendamos que utilice AWS IAM Identity Center. Puede crear usuarios y grupos en IAM Identity Center, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todos sus Cuentas de AWS y aplicaciones. Para obtener información sobre IAM Identity Center, consulte [¿Qué es IAM Identity Center?](#) en el AWS IAM Identity Center Guía del usuario.

Usuarios y grupos de IAM

Un [IAM usuario](#) es una identidad dentro de tu Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos utilizar credenciales temporales en lugar de crear IAM usuarios con credenciales de larga duración, como contraseñas y claves de acceso. Sin embargo, si tiene casos de uso específicos que requieren credenciales a largo plazo con IAM los usuarios, le recomendamos que rote las claves de acceso. Para obtener más información, consulte [Rotar las claves de acceso con regularidad para los casos de uso que requieran credenciales de larga duración](#) en la Guía del IAM usuario.

Un [IAM grupo](#) es una identidad que especifica un conjunto de IAM usuarios. No puede iniciar sesión como grupo. Puede 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 IAM y concederle permisos para administrar IAM los recursos.

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 [Cuándo crear un IAM usuario \(en lugar de un rol\)](#) en la Guía del IAM usuario.

IAM roles

Un [IAM rol](#) es una identidad dentro de tu Cuenta de AWS que tiene permisos específicos. Es similar a un IAM usuario, pero no está asociado a una persona específica. Puede asumir temporalmente un IAM rol en el AWS Management Console [cambiando de rol](#). Puede asumir un rol llamando a un AWS CLI o AWS API operación o mediante una operación personalizada URL. Para obtener más información sobre los métodos de uso de roles, consulte [Uso de IAM roles](#) en la Guía del IAM usuario.

IAM los roles 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 sobre los roles para la federación, consulte [Creación de un rol para un proveedor de identidad externo](#) en la Guía del IAM usuario. Si usa IAM Identity Center, configura un conjunto de permisos. Para controlar a qué pueden acceder sus identidades después de autenticarse, IAM Identity Center correlaciona el conjunto de permisos con un rol en. IAM Para obtener información sobre los conjuntos de permisos, consulte los [conjuntos de permisos](#) en la AWS IAM Identity Center Guía del usuario.
- **Permisos de IAM usuario temporales:** un IAM usuario o rol puede asumir un IAM rol para asumir temporalmente diferentes permisos para una tarea específica.
- **Acceso multicuenta:** puedes usar un IAM rol para permitir que alguien (un responsable de confianza) de una cuenta diferente acceda a los recursos de tu cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunos Servicios de AWS, puede adjuntar una política directamente a un recurso (en lugar de utilizar un rol como proxy). Para saber la diferencia entre las funciones y las políticas basadas en recursos para el acceso multicuenta, consulta el tema sobre el acceso a los [recursos entre cuentas IAM en](#) la Guía del IAM usuario.
- **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 IAM usuario o un rol para realizar acciones en AWS, se le considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del director que llama a un Servicio de AWS, combinado con la solicitud Servicio de AWS para realizar solicitudes a los servicios intermedios. FAS las solicitudes solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS o recursos para completar. En este caso, debe tener permisos para realizar ambas acciones. Para obtener detalles sobre la política a la hora de realizar FAS solicitudes, consulte [Reenviar las sesiones de acceso](#).
- **Función de servicio:** una función de servicio es una [IAM función](#) que un servicio asume para realizar acciones en su nombre. Un IAM administrador puede crear, modificar y eliminar un rol de servicio desde dentro IAM. Para obtener más información, consulte [Crear un rol para delegar permisos a un Servicio de AWS](#) en la Guía del usuario de IAM.

- **Función vinculada a un 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 la función de realizar una acción en su nombre. Los roles vinculados al servicio aparecen en su Cuenta de AWS y son propiedad del servicio. Un IAM administrador puede ver, pero no editar, los permisos de las funciones vinculadas al servicio.
- **Aplicaciones que se ejecutan en Amazon EC2:** puedes usar un IAM rol para administrar las credenciales temporales de las aplicaciones que se ejecutan en una EC2 instancia y se están creando AWS CLI o AWS APIsolicitudes. Esto es preferible a almacenar las claves de acceso en la EC2 instancia. Para asignar un AWS Un rol a una EC2 instancia y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia que se adjunte 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 [Uso de un IAM rol para conceder permisos a aplicaciones que se ejecutan en EC2 instancias de Amazon](#) en la Guía del IAM usuario.

Para saber si se deben usar IAM roles o IAM usuarios, consulte [Cuándo crear un IAM rol \(en lugar de un usuario\)](#) en la Guía del IAM usuario.

Administración de acceso mediante políticas

Usted controla el acceso en AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto en 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 en AWS como JSON documentos. Para obtener más información sobre la estructura y el contenido de los documentos de JSON políticas, consulte [Descripción general de JSON las políticas](#) en la Guía del IAM usuario.

Los administradores pueden utilizar AWS JSONpolíticas 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. Para conceder a los usuarios permiso para realizar acciones en los recursos que necesitan, un IAM administrador puede crear IAM políticas. A continuación, el administrador puede añadir las IAM políticas a las funciones y los usuarios pueden asumir las funciones.

IAMlas políticas definen los permisos para una acción independientemente del método que se utilice 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 su función en AWS Management Console, el AWS CLI, o el AWS API.

Políticas basadas en identidad

Las políticas basadas en la identidad son documentos de política de JSON permisos que se pueden adjuntar a una identidad, como un IAM usuario, 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 información sobre cómo crear una política basada en la identidad, consulte [Creación de IAM políticas](#) en la Guía del usuario. 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 funciones de su Cuenta de AWS. Las políticas gestionadas incluyen AWS las políticas gestionadas y las políticas gestionadas por el cliente. Para saber cómo elegir entre una política gestionada o una política en línea, consulte [Elegir entre políticas gestionadas y políticas integradas en la Guía](#) del IAM usuario.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de JSON política que se adjuntan a un recurso. Algunos ejemplos de políticas basadas en recursos son las políticas de confianza de IAM roles 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 AWS políticas gestionadas desde una política basada IAM 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 JSON políticas.

Amazon S3, AWS WAF, y Amazon VPC son ejemplos de servicios que admiten 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 función avanzada en la que se establecen los permisos máximos que una política basada en la identidad puede conceder a una IAM entidad (IAM usuario o rol). Puede 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 los [límites de los permisos para IAM las entidades](#) en la Guía del IAM usuario.
- **Políticas de control de servicios (SCPs):** SCPs son JSON políticas que especifican los permisos máximos para una organización o unidad organizativa (OU) en AWS Organizations. AWS Organizations es un servicio para agrupar y administrar de forma centralizada múltiples Cuentas de AWS que es propiedad de su empresa. Si habilitas todas las funciones de una organización, puedes aplicar las políticas de control de servicios (SCPs) a cualquiera de tus cuentas o a todas ellas. SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas todas Usuario raíz de la cuenta de AWS. Para obtener más información acerca de Organizations SCPs, consulte [Políticas de control de servicios](#) en AWS Organizations Guía del 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 obtener más información, consulte [las políticas de sesión](#) en la Guía del IAM usuario.

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 AWS determina si se permite una solicitud cuando se

trata de varios tipos de políticas, consulte la [lógica de evaluación de políticas](#) en la Guía del IAM usuario.

Cómo MediaStore funciona AWS Elemental con IAM

Antes de administrar el IAM acceso a MediaStore, infórmate sobre IAM las funciones disponibles para usarlas MediaStore.

IAM funciones que puedes usar con AWS Elemental MediaStore

IAM característica	MediaStore apoyo
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 las políticas)	Parcial
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	Sí
Roles vinculados al servicio	No

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

Políticas basadas en la identidad para MediaStore

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en la identidad son documentos de política de JSON permisos que se pueden adjuntar a una identidad, como un IAM usuario, 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 información sobre cómo crear una política basada en la identidad, consulte [Creación de IAM políticas](#) en la Guía del usuario. IAM

Con las políticas IAM basadas en la identidad, 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 adjunto. Para obtener más información sobre todos los elementos que puede utilizar en una JSON política, consulte la [referencia sobre los elementos de la IAM JSON política](#) en la Guía del IAM usuario.

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 la identidad para Elemental AWS MediaStore](#)

Políticas basadas en recursos incluidas MediaStore

Compatibilidad con las políticas basadas en recursos: sí

Las políticas basadas en recursos son documentos de JSON política que se adjuntan a un recurso. Algunos ejemplos de políticas basadas en recursos son las políticas de confianza de IAM roles 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 una cuenta completa o IAM entidades de otra cuenta como principales en una política basada en 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 IAM administrador

de la cuenta de confianza también debe conceder permiso a la entidad principal (usuario o rol) para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para obtener más información, consulte [Acceso a recursos entre cuentas IAM en](#) la Guía del IAM usuario.

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 política: sí

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

El `Action` elemento de una JSON política describe las acciones que puede utilizar para permitir o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que las asociadas AWS APIoperación. Hay algunas excepciones, como las acciones que solo permiten permisos y que no tienen una operación coincidente. API 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, consulta las [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 la identidad para Elemental AWS MediaStore](#)

Recursos de políticas para MediaStore

Compatibilidad con los recursos de políticas: sí

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

El elemento Resource JSON de política especifica el objeto o los objetos a los que se aplica la acción. Las instrucciones deben contener un elemento Resource o NotResource. Como práctica recomendada, especifique un recurso mediante su [nombre de recurso de Amazon \(ARN\)](#). Puede 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, utilice 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 [los recursos definidos por AWS Elemental MediaStore](#) en la referencia de autorización del servicio. Para saber con qué acciones puedes especificar cada recurso, consulta [las acciones definidas por AWS Elemental MediaStore](#). ARN

El recurso MediaStore contenedor tiene lo 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 Espacios de nombres de servicios](#).

Por ejemplo, para especificar el AwardsShow contenedor en la declaración, utilice lo 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 AWS JSON políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede 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. Puede crear expresiones condicionales que utilicen [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 `Condition` elementos en una declaración o varias claves en un solo `Condition` elemento, AWS los evalúa mediante una AND operación lógica. Si especifica varios valores para una sola 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 puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder a un IAM usuario permiso para acceder a un recurso solo si está etiquetado con su nombre de IAM usuario. Para obtener más información, consulte [los elementos de IAM política: variables y etiquetas](#) en la Guía del IAM usuario.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas AWS claves de condición globales, consulte [AWS claves de contexto de condiciones globales](#) en la Guía IAM del usuario.

Para ver una lista de claves de MediaStore condición, consulte las claves de [condición de AWS Elemental MediaStore](#) en la Referencia de autorización de servicio. Para saber con qué acciones y recursos puedes usar una clave condicionada, consulta [las acciones definidas por AWS Elemental MediaStore](#).

Para ver ejemplos de políticas MediaStore basadas en la identidad, consulte. [Ejemplos de políticas basadas en la identidad para Elemental AWS MediaStore](#)

ACLs en MediaStore

Soportes 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 JSON políticas.

ABAC con MediaStore

Soportes ABAC (etiquetas en las políticas): parciales

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

ABAC es útil en entornos de rápido crecimiento y ayuda en situaciones en las que la administración de políticas se vuelve 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 al respecto ABAC, consulte [¿Qué es? ABAC](#) en la Guía IAM del usuario. Para ver un tutorial con los pasos de configuración ABAC, consulte [Usar el control de acceso basado en atributos \(ABAC\)](#) en la Guía del IAM usuario.

Usar credenciales temporales con MediaStore

Compatibilidad con credenciales temporales: sí

Algunos Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluyendo qué Servicios de AWS trabajen con credenciales temporales, consulte [Servicios de AWS que funcionan IAM](#) en la Guía IAM del usuario.

Está utilizando credenciales temporales si inicia sesión en AWS Management Console utilizando cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accede a AWS mediante el enlace de inicio de sesión único (SSO) de su 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 rol, consulte [Cambiar a un rol \(consola\)](#) en la Guía del IAM usuario.

Puede crear credenciales temporales manualmente mediante el AWS CLI o AWS API. 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 utiliza un IAM usuario o un rol para realizar acciones en AWS, se le considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del director que llama a un Servicio de AWS, combinado con la solicitud Servicio de AWS para realizar solicitudes a los servicios intermedios. FAS las solicitudes solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS o recursos para completar. En este caso, debe tener permisos para realizar ambas acciones. Para obtener detalles sobre la política a la hora de realizar FAS solicitudes, consulte [Reenviar las sesiones de acceso](#).

Roles de servicio para MediaStore

Compatibilidad con roles de servicio: sí

Una función de servicio es una [IAM función](#) que un servicio asume para realizar acciones en su nombre. Un IAM administrador puede crear, modificar y eliminar un rol de servicio desde dentro IAM. Para obtener más información, consulte [Crear un rol para delegar permisos a un Servicio de AWS](#) en la Guía del usuario de IAM.

Warning

Cambiar los permisos de un rol de servicio podría interrumpir 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 la función de realizar una acción en su nombre. Los roles vinculados al servicio aparecen en su Cuenta de AWS y son propiedad del servicio. Un IAM administrador puede ver, pero no editar, los permisos de las funciones vinculadas al servicio.

Para obtener más información sobre la creación o la administración de funciones vinculadas a un servicio, consulte [AWS servicios con los que funcionan. 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 la identidad para Elemental AWS MediaStore

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

Para obtener información sobre cómo crear una política IAM basada en la identidad mediante estos documentos de JSON política de ejemplo, consulte [Creación de IAM políticas](#) en la Guía del IAM usuario.

Para obtener más información sobre las acciones y los tipos de recursos definidos MediaStore, incluido el 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. ARNs

Temas

- [Prácticas recomendadas sobre las políticas](#)
- [Uso de la MediaStore consola](#)
- [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 tu cuenta, acceder a ellos o eliminarlos. Estas acciones pueden suponer costes para su Cuenta de AWS. Al crear o editar políticas basadas en la identidad, siga estas directrices y recomendaciones:

- Comience con AWS políticas gestionadas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice la AWS políticas gestionadas que conceden permisos para muchos casos de uso habituales. Están disponibles en su Cuenta de AWS. Le recomendamos que reduzca aún más los permisos definiendo AWS políticas gestionadas por el cliente que sean específicas para sus casos de uso. Para obtener más información, consulte [AWS políticas gestionadas](#) o [AWS políticas gestionadas para las funciones laborales](#) en la Guía IAM del usuario.
- Aplique permisos con privilegios mínimos: cuando establezca permisos con IAM políticas, 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. Para obtener más información sobre cómo IAM aplicar permisos, consulte [Políticas y permisos IAM en](#) la IAM Guía del usuario.
- Utilice las condiciones en IAM las políticas para restringir aún más el acceso: puede añadir una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de política para especificar que todas las solicitudes deben enviarse mediante SSL. También puede utilizar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de un procedimiento específico Servicio de AWS, como, por ejemplo, AWS CloudFormation. Para obtener más información, consulte [los elementos IAM JSON de la política: Condición](#) en la Guía del IAM usuario.
- Utilice IAM Access Analyzer para validar sus IAM políticas y garantizar permisos seguros y funcionales: IAM Access Analyzer valida las políticas nuevas y existentes para que se ajusten al lenguaje de las políticas (JSON) y IAM a las IAM mejores prácticas. IAM Access Analyzer proporciona más de 100 comprobaciones de políticas y recomendaciones prácticas para ayudarle a crear políticas seguras y funcionales. Para obtener más información, consulte la [validación de políticas de IAM Access Analyzer](#) en la Guía del IAM usuario.
- Requerir autenticación multifactorial (MFA): si tiene un escenario que requiere IAM usuarios o un usuario raíz en su Cuenta de AWS, actívala MFA para mayor seguridad. Para solicitarlo MFA cuando se cancelen API las operaciones, añada MFA condiciones a sus políticas. Para obtener más información, consulte [Configuración del API acceso MFA protegido](#) en la Guía del IAM usuario.

Para obtener más información sobre las prácticas recomendadas IAM, consulte las [prácticas recomendadas de seguridad IAM en](#) la Guía del IAM usuario.

Uso de la MediaStore consola

Para acceder a AWS En MediaStore la consola elemental, debes tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los MediaStore recursos de su Cuenta de AWS. Si creas una política basada en la identidad que sea más restrictiva que los permisos mínimos requeridos, la consola no funcionará según lo previsto para las entidades (usuarios o roles) que cuenten con esa política.

No es necesario conceder permisos mínimos de consola a los usuarios que solo realizan llamadas al AWS CLI o el AWS API. En su lugar, permita el acceso únicamente a las acciones que coincidan con la API operación que están intentando realizar.

Para garantizar que los usuarios y los roles puedan seguir utilizando la MediaStore consola, adjunte también la opción MediaStore *ConsoleAccess* o *ReadOnly* AWS política gestionada a las entidades. Para obtener más información, consulte [Añadir permisos a un usuario](#) en la Guía del IAM usuario.

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo se muestra cómo se puede crear una política que permita a IAM los usuarios ver las políticas integradas y administradas asociadas a su identidad de usuario. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante el AWS CLI o AWS API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "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 MediaStore y 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 mi Cuenta de AWS para acceder 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.

El siguiente ejemplo de error se produce cuando el matoejackson IAM usuario intenta usar la consola para ver detalles sobre un *my-example-widget* recurso ficticio pero no tiene los mediastore:*GetWidget* permisos ficticios.

```
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 necesitas ayuda, ponte en contacto con tu AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

No estoy autorizado a realizar tareas como: PassRole

Si recibes un mensaje de error que indica que no estás autorizado a realizar la `iam:PassRole` acción, debes actualizar tus políticas para que puedas transferirle MediaStore una función.

Alguno 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.

El siguiente ejemplo de error se produce cuando un IAM usuario denominado `marymajor` intenta utilizar la consola para realizar una acción en ella. 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 AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mi Cuenta de AWS para acceder 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 admiten políticas basadas en recursos o listas de control de acceso (ACLs), puedes usar esas políticas para permitir que las personas accedan a tus recursos.

Para 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 sus recursos en Cuentas de AWS que te pertenezca, consulta [Proporcionar acceso a un IAM usuario en otro Cuenta de AWS que le pertenezca](#) en la Guía IAM del usuario.
- Para obtener información sobre cómo proporcionar acceso a sus recursos a terceros Cuentas de AWS, consulte [Proporcionar acceso a Cuentas de AWS propiedad de terceros](#) en la Guía IAM del usuario.
- Para obtener información sobre cómo proporcionar acceso mediante la federación de identidades, consulte [Proporcionar acceso a usuarios autenticados externamente \(federación de identidades\)](#) en la Guía del IAM usuario.
- Para saber la diferencia entre el uso de roles y políticas basadas en recursos para el acceso entre cuentas, consulte el acceso a [recursos entre cuentas IAM en la Guía](#) del usuario. IAM

Inicio de sesión y supervisión AWS Elemental MediaStore

En esta sección se proporciona una descripción general de las opciones para iniciar sesión y supervisar AWS Elemental MediaStore por motivos de seguridad. Para obtener más información sobre el inicio de sesión y la supervisión MediaStore, consulte [Monitorización 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 su AWS soluciones. Debe recopilar datos de monitoreo de todas las partes de su AWS solución para que pueda depurar más fácilmente 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 SNS tema de Amazon o a una política de AWS Auto Scaling. 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 [Registrar llamadas a la API con CloudTrail](#).

AWS Trusted Advisor

Trusted Advisor se basa en las mejores prácticas aprendidas al atender a cientos de miles de AWS clientes. Trusted Advisor inspecciona su AWS entorno 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 AWS los clientes tienen acceso a cinco cheques de Trusted Advisor. Los clientes con un plan de soporte empresarial o empresarial pueden ver todos Trusted Advisor cheques.

Para obtener más información, consulte [AWS Trusted Advisor](#).

Validación de la conformidad de AWS Elemental MediaStore

Para saber si un Servicio de AWS está dentro del ámbito de programas de cumplimiento específicos, consulte [Servicios de AWS dentro del ámbito de aplicación por programa de cumplimiento](#) y elija el programa de cumplimiento que le interese. Para obtener información general, consulte [AWS Programas de cumplimiento](#) .

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 usar Servicios de AWS 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:

- [Guías de inicio rápido sobre seguridad y cumplimiento](#): estas guías de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos para implementar entornos básicos en AWS que se centran en la seguridad y el cumplimiento.
- Diseño de [arquitectura para garantizar HIPAA la seguridad y el cumplimiento en Amazon Web Services](#): en este documento técnico se describe cómo las empresas pueden utilizar AWS para crear aplicaciones aptasHIPAA.

Note

No todos Servicios de AWS son HIPAA aptos. Para obtener más información, consulta la [Referencia de servicios HIPAA aptos](#).

- [AWS Recursos de cumplimiento](#) : esta colección de libros de trabajo y guías puede aplicarse a su sector 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 Servicios de AWS y mapear la guía con los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad de la Industria de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).
- [Evaluación de los recursos con las reglas](#) del AWS Config Guía para desarrolladores: la AWS Config El servicio evalúa en qué medida las configuraciones de sus recursos cumplen con las prácticas internas, las pautas y las regulaciones del sector.
- [AWS Security Hub](#)— Esto Servicio de AWS proporciona una visión completa de su estado de seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar su AWS recursos y para comprobar su conformidad con los estándares y las mejores prácticas 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](#) — Esto Servicio de AWS detecta posibles amenazas para su Cuentas de AWS, cargas de trabajo, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas o maliciosas. GuardDuty puede ayudarle a cumplir diversos requisitos de conformidad, por ejemplo PCIDSS, cumpliendo con los requisitos de detección de intrusiones exigidos por determinados marcos de conformidad.
- [AWS Audit Manager](#)— Esto Servicio de AWS le ayuda a auditar continuamente su AWS uso para simplificar la forma en que gestiona el riesgo y el cumplimiento de las normas y los estándares del sector.

La resiliencia en AWS Elemental MediaStore

La AWS la infraestructura global se construye en torno a Regiones de AWS y zonas de disponibilidad. Regiones de AWS proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas a 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 acerca de Regiones de AWS y zonas de disponibilidad, consulte [AWS Infraestructura global](#).

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

Seguridad de la infraestructura en Elemental AWS MediaStore

Como servicio gestionado, AWS Elemental MediaStore está protegido por AWS seguridad de red global. Para obtener más información AWS servicios de seguridad y cómo AWS protege la infraestructura, consulte [AWS Seguridad en la nube](#). Para diseñar su AWS utilizando las mejores prácticas de seguridad de la infraestructura, consulte el pilar [Protección de la infraestructura](#) en la seguridad AWS Marco bien diseñado.

Usas AWS APIllamadas publicadas para acceder a MediaStore través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Necesitamos TLS 1.2 y recomendamos TLS 1.3.
- Cifre suites con perfecto secreto (PFS), como (Ephemeral Diffie-Hellman) o DHE 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 firmarse con un identificador de clave de acceso y una clave de acceso secreta asociada a un director. IAM O bien, puede utilizar la [AWS Security Token Service](#) (AWS STS) para generar credenciales de seguridad temporales para firmar las 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 puede 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 puede 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 le ayudan a proteger sus datos para todos los servicios cuyos directores de servicio tengan acceso a los recursos de su cuenta.

Recomendamos usar las claves de contexto de condición [aws:SourceAccount](#) global [aws:SourceArn](#) las claves contextuales en las políticas de recursos para limitar los permisos que AWS Elemental MediaStore otorga a otro servicio al recurso. Utilice `aws:SourceArn` si desea que solo se asocie un recurso al acceso entre servicios. Utilice `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 confuso problema de los diputados es utilizar la clave de contexto ARN de la condición `aws:SourceArn` global con todo el recurso. Si no conoce la totalidad ARN del recurso o si está especificando varios recursos, utilice la clave de condición del contexto `aws:SourceArn` global con caracteres comodín (*) para las partes desconocidas del ARN. Por ejemplo, `arn:aws:servicename::123456789012::*`.

Si el `aws:SourceArn` valor no contiene el ID de la cuenta, como un bucket de Amazon S3 ARN, debe usar 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"
  }
}
}
```

Monitorización y etiquetado en AWS Elemental MediaStore

El monitoreo es una parte importante a la hora de mantener la fiabilidad, la disponibilidad y el rendimiento de AWS Elemental MediaStore y AWS las demás soluciones de. AWS ofrece las siguientes herramientas de monitoreo para ver MediaStore, informar cuando algo no funciona y realizar acciones automáticas cuando corresponda:

- AWS CloudTrail captura llamadas a la API y eventos relacionados efectuados por su cuenta de AWS o en su nombre, y entrega los archivos de registro al bucket de Amazon S3 que se haya especificado. También pueden identificar qué usuarios y cuentas llamaron a AWS, la dirección IP de origen de las llamadas y el momento en que se hicieron. Para obtener más información, consulte la [Guía del usuario de AWS CloudTrail](#).
- Amazon CloudWatch monitorea AWS los recursos de y las aplicaciones que ejecuta 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, puede hacer que realice CloudWatch un seguimiento del uso de la CPU u otras métricas de las instancias de Amazon EC2 y lanzar nuevas instancias automáticamente cuando sea necesario. Para obtener más información, consulte la [Guía del CloudWatch usuario de Amazon](#).
- Amazon CloudWatch Events proporciona un flujo de eventos de sistema que describen cambios en los AWS recursos de. Normalmente AWS los servicios de enviar notificaciones de CloudWatch eventos a Events en cuestión de segundos, pero a veces pueden tardar un minuto o más. CloudWatch Events habilita la informática basada en eventos automatizada, para que pueda escribir reglas que vigilan determinados eventos y desencadenan acciones automatizadas en otros AWS servicios de cuando estos eventos se producen. Para obtener más información, consulte la [Guía del usuario de Amazon CloudWatch Events](#).
- Amazon CloudWatch Logs le permite monitorear, almacenar y tener acceso a los archivos de registro desde instancias de Amazon EC2 u otras fuentes. CloudTrail CloudWatch Los registros pueden monitorear información en los archivos de registro y enviarle una notificación cuando se llega a determinados umbrales. También se pueden archivar los datos de los registros en un almacenamiento de larga duración. Para obtener más información, consulte la [Guía del usuario de Amazon CloudWatch Logs](#).

También puede asignar metadatos a los MediaStore contenedores de 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 utilizar etiquetas para organizar sus recursos de AWS en la consola de administración de AWS, crear informes de uso y facturación en todos sus recursos de AWS y filtrar los recursos durante las actividades de automatización de la infraestructura.

Temas

- [Registro de llamadas a la MediaStore API de AWS Elemental con AWS CloudTrail](#)
- [Supervisión de AWS Elemental MediaStore con Amazon CloudWatch](#)
- [Etiquetado de los recursos de AWS Elemental MediaStore](#)

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

AWS Elemental MediaStore se integra con AWS CloudTrail, un servicio que proporciona un registro de las acciones llevadas a cabo por un usuario, un rol o un AWS servicio de en MediaStore.

CloudTrail captura un subconjunto de llamadas a la API de MediaStore como eventos, incluidas las llamadas procedentes de la MediaStore consola de y las llamadas de código a la MediaStore API de. Si crea un registro de seguimiento, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos para MediaStore. Si no configura un registro de seguimiento, puede ver los eventos más recientes de la CloudTrail consola de en el Event history (Historial de eventos). Mediante la información que recopila CloudTrail, se puede determinar la solicitud que se envió a MediaStore, la dirección IP desde la que se realizó la solicitud, quién la realizó, cuándo la realizó y detalles.

Para obtener más información CloudTrail, incluso cómo configurarlo y habilitarlo, consulte la [Guía del AWS CloudTrail usuario de](#).

Temas

- [MediaStore Información de AWS Elemental en CloudTrail](#)
- [Ejemplo: entradas de archivos de MediaStore registro de AWS Elemental](#)

MediaStore Información de AWS Elemental en CloudTrail

CloudTrail se habilita en su AWS cuenta de al crearla. Cuando se produce una actividad de eventos compatible en AWS Elemental MediaStore, la actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicios de en Event history (Historial de eventos). Puede ver, buscar

y descargar los últimos eventos de la cuenta de AWS. Para obtener más información, consulte [Visualización de eventos con el historial de CloudTrail eventos de](#).

Para mantener un registro continuo de eventos en la cuenta de AWS, incluidos los eventos de MediaStore, cree un registro de seguimiento. Un registro de seguimiento CloudTrail permite 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. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registro al bucket de Amazon S3 especificado. También es posible configurar otros AWS servicios de para analizar en profundidad y actuar en función de los datos de eventos recopilados en los CloudTrail registros de. Para obtener más información, consulte los siguientes temas:

- [Introducción a la creación de registros de seguimiento](#)
- [CloudTrail Integraciones y servicios compatibles](#)
- [Configuración de notificaciones de Amazon SNS para CloudTrail](#)
- [Recepción de archivos de CloudTrail registro de varias regiones](#) y [Recepción de archivos de CloudTrail registro de varias cuentas](#)

AWS Elemental MediaStore admite el registro de las siguientes operaciones como eventos en 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 de
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado
- Si la solicitud la realizó otro servicio de AWS.

Para obtener más información, consulte el [elemento `userIdentity` de CloudTrail](#).

Ejemplo: entradas de archivos de MediaStore registro de AWS Elemental

Un registro de seguimiento es una configuración que permite entregar eventos como archivos de registro al bucket de Amazon S3 que especifique. CloudTrail los archivos de registro pueden contener una o varias entradas de registro. Un evento representa una solicitud específica realizada desde un origen y contiene información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrail los archivos de registro no son un rastro de la pila ordenado de las llamadas a las API públicas, por lo que no aparecen en ningún orden específico.

En el ejemplo siguiente, se muestra una entrada de CloudTrail registro de que ilustra 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":
    "MNCTGH4HRQJ27GRMBVDPIVHEP4L02BN6MUVHBCPSHOAWNSOKSXC024B2UE0BBND5DONRXTMFK3TOJ4G7AHWMESI",
    "eventID": "7085b140-fb2c-409b-a329-f567912d704c",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }

```

Supervisión de AWS Elemental MediaStore con Amazon CloudWatch

Puede monitorear el MediaStore uso de AWS Elemental CloudWatch, que recopila datos sin procesar y los procesa en métricas legibles. CloudWatch mantiene las estadísticas durante 15 meses, lo que le permite tener acceso a información histórica y disponer de una mejor perspectiva sobre el rendimiento de su aplicación web o servicio. También puede establecer alarmas que vigilen determinados umbrales y enviar notificaciones o realizar acciones cuando se cumplan dichos umbrales. Para obtener más información, consulte la [Guía del CloudWatch usuario de Amazon](#).

AWS ofrece las siguientes herramientas de monitoreo para ver MediaStore, informar cuando algo no funciona y realizar acciones automáticas cuando corresponda:

- Amazon CloudWatch Logs le permite monitorear, almacenar y tener acceso a los archivos de registro desde AWS servicios como AWS Elemental MediaStore. Puede utilizar CloudWatch Registros para monitorear aplicaciones y sistemas mediante datos de registro. Por ejemplo,

CloudWatch Logs puede realizar el seguimiento de los errores presentes en los registros de las aplicaciones y enviarle una notificación cuando la tasa de errores supere el umbral que se especifique. CloudWatch Logs utiliza los datos de registro de para el monitoreo, 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 está buscando, CloudWatch Logs notifica 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. Normalmente AWS los servicios de enviar notificaciones de CloudWatch eventos a Events en cuestión de segundos, pero a veces pueden tardar un minuto o más. Puede configurar reglas para que coincidan los eventos (como una `DeleteObject` solicitud) y dirigirlos a una o más secuencias o funciones de destino. CloudWatch Events toma conocimiento de los cambios operativos a medida que se producen. Además, CloudWatch Events responde a estos cambios operativos y toma medidas correctoras según sea necesario, enviando mensajes para responder al entorno, activando funciones, realizando cambios y captando 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 mejor su base de clientes y entender su MediaStore factura de. 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 registros 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 registro con un nombre como `/aws/mediastore/MyContainerName`. Puede definir grupos de registros 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 registros.

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 MediaStore acceso a AWS Elemental 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 registro y flujos de registro para cualquier contenedor de cualquier región de su AWS cuenta.

4. Elija Review policy (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, seleccione Create role (Crear rol).
7. Elija el tipo de rol Another AWS account (Otra cuenta de AWS).
8. En Account ID (ID de cuenta), escriba el ID de su cuenta de AWS.
9. Elija Next: Permissions (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 Role name (Nombre de rol), escriba **MediaStoreAccessLogs** y luego elija Create role (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 Edit trust relationship (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": [  
    {  
      "Principal": {  
        "Service": "mediastore.amazonaws.com"  
      },  
      "Action": "iam:CreateRole",  
      "Effect": "Allow",  
      "Resource": "*"
    }
  ]
}
```

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "mediastore.amazonaws.com"
  },
  "Action": "sts:AssumeRole",
  "Condition": {}
}
]
```

18. Elija Update Trust Policy (Actualizar política de confianza).

Habilitar el registro de acceso para un contenedor

De forma predeterminada, AWS Elemental MediaStore no recopila registros de acceso. Al habilitar el registro de acceso en un contenedor, MediaStore proporciona registros de acceso para los objetos almacenados en dicho contenedor a Amazon 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 la AWS CLI, utilice el comando `start-access-logging`:

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

```
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 a AWS Elemental no aparecen en Amazon CloudWatch, consulte la siguiente tabla para conocer las posibles causas y las soluciones.

Note

Asegúrese de habilitar AWS CloudTrail Logs para ayudarle con el proceso de resolución 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 “Configuración de permisos para CloudWatch” .
Envío una solicitud de API <code>DescribeContainer</code> , pero la respuesta muestra que el parámetro <code>AccessLog</code>	El rol de IAM no existe o tiene el nombre, permisos o	Cree un rol con el nombre, permisos y política de confianza correctos. Consulte

Síntoma	El Problema Might Be...	Pruebe esto...
<p>gingEnabled tiene un valor de False. Además, no puede ver ningún evento de CloudTrail para que el rol MediaStoreAccessLogs realice una llamada a DescribeLogGroup , CreateLogGroup , DescribeLogStream o CreateLogStream satisfactoria.</p>	<p>política de confianza incorrectos.</p>	<p>the section called “Configuración de permisos para CloudWatch”.</p>
<p>En la CloudTrail consola, aparece un evento con un error de acceso denegado relacionado con elMediaStoreAccessLogs rol. El CloudTrail evento puede incluir líneas como las siguientes:</p> <pre>"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:",</pre>	<p>El rol de IAM no tiene los permisos correctos para AWS Elemental MediaStore.</p>	<p>Habilite los registros de acceso para el contenedor. Consulte the section called “Habilitar el registro de acceso”.</p> <p>Actualice el rol de IAM para que tenga los permisos y la política de confianza correctos . Consulte the section called “Configuración de permisos para CloudWatch”.</p>

Síntoma	El Problema Might Be...	Pruebe esto...
No puede ver ningún registro de un contenedor completo o de contenidos.	Es posible que su cuenta haya superado la CloudWatch cuota de grupos de registro por cuenta y región. Consulte las cuotas de los grupos de registro en la Guía del usuario de Amazon CloudWatch Logs .	En la CloudWatch consola, determine si su cuenta ha alcanzado la CloudWatch cuota de grupos de registro. Si es necesario, solicite un aumento de cuota .
Ve algunos registros iniciados CloudWatch, pero no todos los registros que espera ver.	Es posible que tu cuenta haya superado la CloudWatch cuota de transacciones por segundo por cuenta y región. Consulte las cuotas PutLogEvents en la Guía del usuario de Amazon CloudWatch Logs .	Solicita un aumento de cuota para CloudWatch las transacciones por segundo, 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 ID de la cuenta de AWS utilizada 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 de MediaStore 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 [transient data rule](#) política del ciclo de vida 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 tiene una regla de datos transitorios que se aplique al objeto, o si no se aplica ninguna política de ciclo de vida 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

Una cadena generada por AWS Elemental MediaStore para identificar de forma inequívoca cada solicitud.

Fuente

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 tarda en procesar su 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 AWS los servicios de y responder automáticamente a eventos del sistema, como problemas de disponibilidad de aplicaciones o 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

Normalmente AWS los servicios de enviar 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 consecutivos 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 cómo suscribirse a estos eventos, consulte [Amazon CloudWatch](#).

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

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

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

- Activación de una función Lambda siempre que se cree un contenedor
- Notificar un tema de Amazon SNS cuando se elimina un objeto

Para obtener más información, consulte la [Guía del usuario de Amazon CloudWatch Events](#).

Temas

- [Evento de cambio de estado de MediaStore objetos de AWS Elemental](#)
- [Evento de cambio de estado del MediaStore contenedor de AWS Elemental](#)

Evento de cambio de estado de MediaStore objetos 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 cómo suscribirse a este evento, consulte [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 del 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 cómo suscribirse a este evento, consulte [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"  
}  
}
```

Supervisión de AWS Elemental MediaStore con CloudWatch métricas de Amazon

Puede monitorear el MediaStore uso de AWS Elemental CloudWatch, que recopila datos sin procesar y los procesa en métricas legibles. CloudWatch mantiene las estadísticas de durante 15 meses, lo que le permite tener acceso a información histórica y disponer de una mejor perspectiva sobre el rendimiento de su aplicación web o servicio. También puede establecer alarmas que vigilen determinados umbrales y enviar notificaciones o realizar acciones cuando se cumplan dichos umbrales. Para obtener más información, consulte 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 umbral determinado.

Para consultar las métricas desde la CloudWatch consola de

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 la CloudWatch consola AWS Management Console y ábrala en <https://console.aws.amazon.com/cloudwatch/>.
2. En el panel de navegación, seleccione Metrics (Métricas).
3. En Todas las métricas, elija el espacio de MediaStore nombres AWS/.
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 métricas mediante la AWS CLI

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

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

MediaStore Métricas de AWS Elemental

La siguiente tabla muestra las métricas a las que se MediaStore envía AWS Elemental CloudWatch.

Note

Para ver las métricas, debes [añadir una política de métricas](#) al contenedor que permita MediaStore enviar métricas a Amazon CloudWatch.

Métrica	Descripción
RequestCount	<p>Número total de solicitudes HTTP realizadas a un contenedor de MediaStore, separadas por el tipo de operación (Put, Get, Delete, Describe, List).</p> <p>Unidades: recuento</p> <p>Dimensiones válidas:</p> <ul style="list-style-type: none"> • Nombre de contenedor • Nombre del grupo de objetos • Tipo de solicitud <p>Estadísticas válidas: Sum</p>
4xxErrorCount	<p>La cantidad de solicitudes HTTP realizadas a MediaStore esa dirección provocó un error 4xx.</p> <p>Unidades: recuento</p> <p>Dimensiones válidas:</p> <ul style="list-style-type: none"> • Nombre de contenedor • Nombre del grupo de objetos • Tipo de solicitud

Métrica	Descripción
<p>5xxErrorCount</p>	<p>Estadísticas válidas: Sum</p> <p>La cantidad de solicitudes HTTP realizadas a MediaStore esa dirección provocó un error 5xx.</p> <p>Unidades: recuento</p> <p>Dimensiones válidas:</p> <ul style="list-style-type: none"> • Nombre de contenedor • Nombre del grupo de objetos • Tipo de solicitud <p>Estadísticas válidas: Sum</p>
<p>BytesUploaded</p>	<p>El número de bytes cargados para las solicitudes realizadas a un contenedor de MediaStore en las que la solicitud incluye un cuerpo.</p> <p>Unidades: bytes</p> <p>Dimensiones válidas:</p> <ul style="list-style-type: none"> • Nombre de contenedor • Nombre del grupo de objetos <p>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.</p>

Métrica	Descripción
BytesDownloaded	<p>Número de bytes descargados para las solicitudes realizadas a un contenedor de MediaStore en las que la respuesta contiene un cuerpo.</p> <p>Unidades: bytes</p> <p>Dimensiones válidas:</p> <ul style="list-style-type: none">• Nombre de contenedor• Nombre del grupo de objetos <p>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.</p>
TotalTime	<p>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 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.</p> <p>Unidades: milisegundos</p> <p>Dimensiones válidas:</p> <ul style="list-style-type: none">• Nombre de contenedor• Nombre del grupo de objetos• Tipo de solicitud <p>Estadísticas válidas: Average, Min (igual que P0,0), Max (igual que p100), cualquier percentil entre p0,0 y p100</p>

Métrica	Descripción
TurnaroundTime	<p>La cantidad de milisegundos que se MediaStore tarda en procesar su solicitud. Este valor se mide desde el momento en que se MediaStore recibe el último byte de la solicitud hasta el momento en que se envía el primer byte de la respuesta.</p> <p>Unidades: milisegundos</p> <p>Dimensiones válidas:</p> <ul style="list-style-type: none"> • Nombre de contenedor • Nombre del grupo de objetos • Tipo de solicitud <p>Estadísticas válidas: Average, Min (igual que P0,0), Max (igual que p100), cualquier percentil entre p0,0 y p100</p>
ThrottleCount	<p>Se limitó el número de solicitudes HTTP realizadas a MediaStore esa dirección.</p> <p>Unidades: recuento</p> <p>Dimensiones válidas:</p> <ul style="list-style-type: none"> • Nombre de contenedor • Nombre del grupo de objetos • Tipo de solicitud <p>Estadísticas válidas: Sum</p>

Etiquetado de los recursos de AWS Elemental MediaStore

Una etiqueta es un atributo personalizado que usted o AWS asignan a un recurso de AWS. 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:

- Identificar y organizar sus recursos de AWS. 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 MediaStore *contenedor* de AWS Elemental que a una AWS Elemental MediaLive entrada.
- Realizar un seguimiento de los costos de AWS. Estas etiquetas se activan en el panel de AWS Billing and Cost Management. 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:

- *contenedor*

Para obtener información acerca de cómo añadir y administrar etiquetas, consulte [Administración de 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). Los recursos de Amazon EC2 admiten cualquier carácter.
- 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 prefijo `aws:` está prohibido para las etiquetas; está reservado para su uso por parte de AWS. 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.

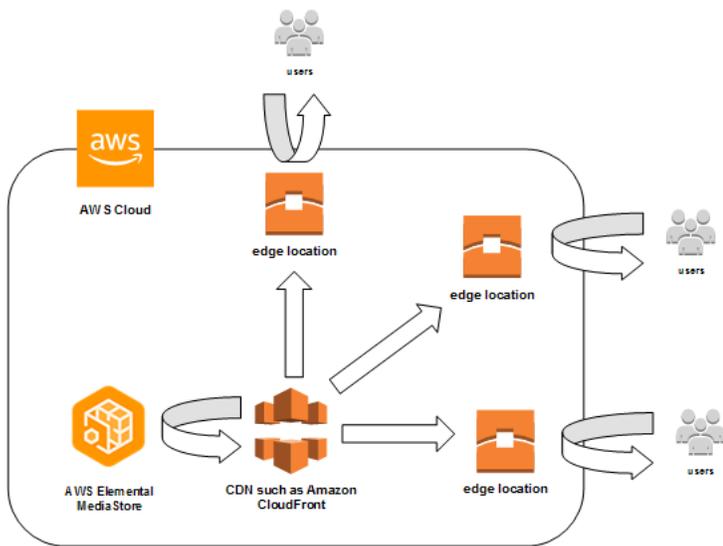
Administración de etiquetas

Las etiquetas se componen de las propiedades `Value` y `Key` 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](#)

Uso de redes de entrega de contenido (CDN)

Puede utilizar 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 periférica, la CDN lo recupera de su origen (como su MediaStore contenedor) y lo distribuye al usuario.



Temas

- [Permitir que Amazon acceda CloudFront a su MediaStore contenedor de AWS Elemental](#)
- [Interacción MediaStore de AWS Elemental con cachés HTTP](#)

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

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

- [Uso del control de acceso a origen \(OAC\)](#)- (Recomendado) Utilice esta opción si Región de AWS admite la función OAC de CloudFront.

- [Uso de secretos compartidos](#)- Utilice esta opción si Región de AWS no admite la función OAC de CloudFront.

Uso del control de acceso a origen (OAC)

Puede utilizar la función Origin Access Control (OAC) de Amazon CloudFront para proteger AWS Elemental MediaStore Origins con una seguridad mejorada. Puede habilitar la [versión 4 de AWS Signature \(Sigv4\)](#) en CloudFront las solicitudes de MediaStore orígenes y establecer cuándo y si CloudFront debe firmar las solicitudes. Puede acceder a la función OAC CloudFront a través de la consola, las API, el SDK o la CLI, y su uso no conlleva cargos adicionales.

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 no Región de AWS 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 Región de AWS la admite. Los siguientes procedimientos requieren que configure MediaStore y CloudFront con secretos compartidos para restringir el acceso a MediaStore los contenedores. Para seguir las mejores prácticas de seguridad, esta configuración manual requiere una rotación periódica de los secretos. Con OAC en MediaStore los orígenes, puede indicar CloudFront que firme las solicitudes mediante SigV4 y reenviarlas MediaStore para que coincidan con las firmas, lo que elimina la necesidad de usar y rotar 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 sea más CloudFront sencilla MediaStore y segura.

Para permitir CloudFront el acceso a tu 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 otorgue acceso de lectura o superior a Amazon CloudFront.

Example

La siguiente política de ejemplo, que es similar a la política de ejemplo para el [acceso público de lectura a través de HTTPS](#), cumple con estos requisitos porque permite `GetObjectDescribeObject` recibir órdenes de cualquier persona que envíe solicitudes a tu dominio a través de HTTPS. Además, la siguiente política de ejemplo protege mejor el flujo de trabajo porque solo permite el CloudFront acceso a MediaStore los objetos cuando la solicitud se realiza 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 extremo 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, consulte 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.

Después de adjuntar la política a sus MediaStore contenedores, debe configurarlo CloudFront para usar solo conexiones HTTPS para las solicitudes de origen y también agregar un encabezado personalizado con el valor secreto correcto.

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

1. Abra Abra Abra CloudFront la de.
2. En la página Orígenes, elige tu MediaStore origen.
3. Elija Edit (Editar).
4. Elija HTTPS solo para el protocolo.
5. En la sección Agregar encabezado personalizado, selecciona Agregar encabezado.
6. Para el nombre, elija Referer. Para el valor, utilice la misma <secretValue>cadena que utilizó en su política de contenedores.
7. Elige Guardar y deja que se implementen los cambios.

Interacción MediaStore de AWS Elemental con cachés HTTP

AWS Elemental MediaStore almacena los objetos para que las redes de entrega de contenido (CDN) como Amazon puedan almacenarlos en caché de manera correcta y eficiente CloudFront. 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 HTTP 1.1 se encuentran en la [sección 13 de 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 CDN y los navegadores web que cumplen con los estándares utilizan esta etiqueta como clave para almacenar en caché el objeto. MediaStore genera automáticamente un ETag para cada objeto cuando se carga. Puede [ver los detalles de un objeto](#) para determinar su valor de ETag.
- **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 a 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 establece 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, como se describe en el [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 CDN

Las CDN que ofrecen contenido en nombre de MediaStore pueden procesar las solicitudes condicionales devolviéndolas `304 Not Modified`, tal como se describe en la [sección 4.1 del RFC7232](#). 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.

Los CDN (y otras cachés que son compatibles con HTTP/1.1) basan estas decisiones en los encabezados `ETag` y `Cache-Control` que los servidores de origen reenvían. Para controlar la frecuencia con la que las CDN consultan a los servidores de MediaStore origen para obtener actualizaciones de los objetos recuperados repetidamente, defina los `Cache-Control` encabezados de esos objetos cuando los cargue 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 uno SDK proporciona API ejemplos de código y documentación que facilitan a los desarrolladores la creación de aplicaciones en su idioma preferido.

SDKdocumentación	Ejemplos de código
AWS SDK for C++	AWS SDK for C++ ejemplos de código
AWS CLI	AWS CLI ejemplos de código
AWS SDK for Go	AWS SDK for Go ejemplos de código
AWS SDK for Java	AWS SDK for Java ejemplos de código
AWS SDK for JavaScript	AWS SDK for JavaScript ejemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin ejemplos de código
AWS SDK for .NET	AWS SDK for .NET ejemplos de código
AWS SDK for PHP	AWS SDK for PHP ejemplos de código
AWS Tools for PowerShell	Herramientas para ejemplos PowerShell de código
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) ejemplos de código
AWS SDK for Ruby	AWS SDK for 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

 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 AWS kit de desarrollo de 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 funciones de servicio individuales, es posible ver las acciones en contexto en los escenarios relacionados.

Para obtener una lista completa de AWS SDKguías para desarrolladores y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo empezar y detalles sobre SDK las versiones anteriores.

Ejemplos de código

- [Ejemplos básicos de MediaStore uso AWS SDKs](#)
 - [Acciones para MediaStore usar AWS SDKs](#)
 - [CreateContainerÚselo con un AWS SDKo CLI](#)
 - [DeleteContainerÚselo con un AWS SDKo CLI](#)
 - [DeleteObjectÚselo con un AWS SDKo CLI](#)
 - [DescribeContainerÚselo con un AWS SDKo CLI](#)
 - [GetObjectÚselo con un AWS SDKo CLI](#)
 - [ListContainersÚselo con un AWS SDKo CLI](#)
 - [PutObjectÚselo con un AWS SDKo 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](#)
 - [CreateContainerÚselo con un AWS SDKo CLI](#)
 - [DeleteContainerÚselo con un AWS SDKo CLI](#)
 - [DeleteObjectÚselo con un AWS SDKo CLI](#)

- [DescribeContainerÚselo con un AWS SDKo CLI](#)
- [GetObjectÚselo con un AWS SDKo CLI](#)
- [ListContainersÚselo con un AWS SDKo CLI](#)
- [PutObjectÚselo con un AWS SDKo 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 obtener una lista completa, consulta la [AWS Elemental MediaStore APIReferencia](#).

Ejemplos

- [CreateContainerÚselo con un AWS SDKo CLI](#)
- [DeleteContainerÚselo con un AWS SDKo CLI](#)
- [DeleteObjectÚselo con un AWS SDKo CLI](#)
- [DescribeContainerÚselo con un AWS SDKo CLI](#)
- [GetObjectÚselo con un AWS SDKo CLI](#)
- [ListContainersÚselo con un AWS SDKo CLI](#)
- [PutObjectÚselo con un AWS SDKo CLI](#)

CreateContainerÚselo con un AWS SDKo CLI

En los siguientes ejemplos de código, se muestra cómo utilizar CreateContainer.

CLI

AWS CLI

Para crear un contenedor

En el siguiente create-container ejemplo, se crea un contenedor nuevo y 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 el AWS Guía MediaStore del usuario de Elemental.

- Para API obtener más información, consulte [CreateContainer](#) en AWS CLI Referencia de comandos.

Java**SDK para Java 2.x****Note**

Hay más información. GitHub Consulta el ejemplo completo y aprende a configurarlo y a ejecutarlo en [AWS Repositorio de ejemplos de código](#).

```
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 API obtener más información, consulte [CreateContainer](#) en AWS SDK for Java 2.x APIReferencia.

Para obtener una lista completa de AWS SDKguías para desarrolladores y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo empezar y detalles sobre SDK las versiones anteriores.

DeleteContainerÚselo con un AWS SDKo CLI

En los siguientes ejemplos de código, se muestra cómo utilizar DeleteContainer.

CLI

AWS CLI

Para eliminar un contenedor

En el siguiente delete-container ejemplo, 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 el AWS Guía MediaStore del usuario de Elemental.

- Para API obtener más información, consulte [DeleteContainer](#) en AWS CLI Referencia de comandos.

Java

SDK para Java 2.x

Note

Hay más información. GitHub Consulta el ejemplo completo y aprende a configurarlo y a ejecutarlo en [AWS Repositorio de ejemplos de código](#).

```
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 API obtener más información, consulte [DeleteContainer](#) en AWS SDK for Java 2.x API Referencia.

Para obtener una lista completa de AWS SDK guías para desarrolladores y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo empezar y detalles sobre SDK las versiones anteriores.

DeleteObject Úselo con un AWS SDK o CLI

El siguiente ejemplo de código muestra cómo usarlo `DeleteObject`.

Java

SDK para Java 2.x

Note

Hay más información. GitHub Consulta el ejemplo completo y aprende a configurarlo y a ejecutarlo en [AWS Repositorio de ejemplos de código](#).

```
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 API obtener más información, consulte [DeleteObject](#) en AWS SDK for Java 2.x API Referencia.

Para obtener una lista completa de AWS SDK guías para desarrolladores y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo empezar y detalles sobre SDK las versiones anteriores.

DescribeContainer Úselo con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `DescribeContainer`.

CLI

AWS CLI

Para ver los detalles de un contenedor

En el siguiente describe-container ejemplo, 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://aaabbbcccddee.data.mediastore.us-  
west-2.amazonaws.com"  
  }  
}
```

Para obtener más información, consulte [Visualización de los detalles de un contenedor](#) en el AWS Guía MediaStore del usuario de Elemental.

- Para API obtener más información, consulte [DescribeContainer](#) en AWS CLI Referencia de comandos.

Java

SDK para Java 2.x

Note

Hay más información. GitHub Consulta el ejemplo completo y aprende a configurarlo y a ejecutarlo en [AWS Repositorio de ejemplos de código](#).

```
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 API obtener más información, consulte [DescribeContainer](#) en AWS SDK for Java 2.x API Referencia.

Para obtener una lista completa de AWS SDK guías para desarrolladores y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo empezar y detalles sobre SDK las versiones anteriores.

GetObject Úselo con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar GetObject.

CLI

AWS CLI

Para descargar un objeto

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

```
aws mediastore-data get-object \
```


Java

SDK para Java 2.x

 Note

Hay más información. GitHub Consulta el ejemplo completo y aprende a configurarlo y a ejecutarlo en [AWS Repositorio de ejemplos de código](#).

```
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 outputStream = new FileOutputStream(targetFile);
        outputStream.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 API obtener más información, consulte [GetObject](#) en AWS SDK for Java 2.x API Referencia.

Para obtener una lista completa de AWS SDK guías para desarrolladores y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo empezar y detalles sobre SDK las versiones anteriores.

ListContainers Úselo con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar ListContainers.

CLI

AWS CLI

Para ver una lista de contenedores

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

```
aws mediastore list-containers
```

Salida:

```
{
  "Containers": [
    {
      "CreationTime": 1505317931,
      "Endpoint": "https://aaabbbcccddee.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://ffffggghhhiiijj.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 el AWS Guía MediaStore del usuario de Elemental.

- Para API obtener más información, consulte [ListContainers](#) en AWS CLI Referencia de comandos.

Java

SDK para Java 2.x

Note

Hay más información. GitHub Consulta el ejemplo completo y aprende a configurarlo y a ejecutarlo en [AWS Repositorio de ejemplos de código](#).

```
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 API obtener más información, consulte [ListContainers](#) en AWS SDK for Java 2.x API Referencia.

Para obtener una lista completa de AWS SDK guías para desarrolladores y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo empezar y detalles sobre SDK las versiones anteriores.

PutObject Úselo con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar PutObject.

CLI

AWS CLI

Para cargar un objeto

En el siguiente put-object ejemplo, 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 la carpeta.

```

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 el AWS Guía del MediaStore usuario de Elemental.

- Para API obtener más información, consulte [PutObject](#) en AWS CLI Referencia de comandos.

Java

SDK para Java 2.x

 Note

Hay más información. GitHub Consulta el ejemplo completo y aprende a configurarlo y a ejecutarlo en [AWS Repositorio de ejemplos de código](#).

```
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 API obtener más información, consulte [PutObject](#) en AWS SDK for Java 2.x APIReferencia.

Para obtener una lista completa de AWS SDK guías para desarrolladores y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo empezar y detalles sobre SDK las 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 cuotas de servicio 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 los contenedores individuales de su cuenta, póngase en contacto con el Support de AWS o con su administrador de cuentas. Esta opción puede ayudarte a dividir los límites a nivel de cuenta entre tus contenedores para evitar que un contenedor consuma toda tu 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
Objetos	Sin límite	Puede cargar tantos objetos como desee en una carpeta o contenedor de su cuenta.
Tasa de solicitudes de API DeleteObject	100	<p>Número máximo de solicitudes de operación que puede realizar por segundo. Las solicitudes adicionales se limitan.</p> <p>Puede solicitar un aumento de cuota.</p>
Tasa de solicitudes de API DescribeObject	1 000	<p>Número máximo de solicitudes de operación que puede realizar por segundo. Las solicitudes adicionales se limitan.</p> <p>Puede solicitar un aumento de cuota.</p>
Tasa de solicitudes de GetObject API para la disponibilidad de carga estándar	1 000	<p>Número máximo de solicitudes de operación que puede realizar por segundo. Las solicitudes adicionales se limitan.</p> <p>Puede solicitar un aumento de cuota.</p>
Tasa de solicitudes de GetObject API para la disponibilidad de carga en streaming	25	<p>Número máximo de solicitudes de operación que puede realizar por segundo. Las solicitudes adicionales se limitan.</p> <p>Puede solicitar un aumento de cuota.</p>
Tasa de solicitudes de API ListItems	5	<p>Número máximo de solicitudes de operación que puede realizar por segundo. Las solicitudes adicionales se limitan.</p> <p>Puede solicitar un aumento de cuota.</p>

Recurso u operación	Cuota predeterminada	Comentarios
Tasa de solicitudes de PutObject API para la codificación de transferencia fragmentada (también conocida como disponibilidad de carga en streaming)	10	<p>Número máximo de solicitudes de operación que puede realizar por segundo. Las solicitudes adicionales se limitan.</p> <p>Puede solicitar un aumento de cuota. En la solicitud, especifique el TPS solicitado y el tamaño de objeto medio.</p>
Tasa de solicitudes de PutObject API para la disponibilidad de carga estándar	100	<p>Número máximo de solicitudes de operación que puede realizar por segundo. Las solicitudes adicionales se limitan.</p> <p>Puede solicitar un aumento de cuota. En la solicitud, especifique el TPS solicitado y el tamaño de objeto medio.</p>
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 tabla siguiente, se enumeran los recursos relacionados que le resultarán muy útiles cuando trabaje con AWS Elemental MediaStore.

- [Clases y talleres](#): enlaces a cursos basados en roles y especializados y también a laboratorios autoguiados para ayudarlo a AWS mejorar sus capacidades sobre y obtener experiencia práctica.
- [AWS Centro para desarrolladores](#) de: explore los tutoriales, descargue herramientas y obtenga información sobre los eventos AWS para desarrolladores.
- [AWS Developer Tools \(Herramientas para desarrolladores de\)](#): enlaces a herramientas para desarrolladores, SDK, conjuntos de herramientas de IDE y herramientas de línea de comandos para desarrollar y administrar AWS aplicaciones de.
- [Centro de recursos de introducción](#): aprenda a configurar su () Cuenta de AWS, únase a la AWS comunidad y lance su primera aplicación.
- [Tutoriales prácticos](#): comience con step-by-step tutoriales antes de lanzar su primera aplicación en AWS.
- [AWS Documentos](#) técnicos de: enlaces a una lista completa de AWS documentos técnicos de que tratan una gran variedad de temas técnicos, como arquitecturas, seguridad y economía de la nube, escritos por arquitectos de AWS soluciones de o expertos técnicos.
- [AWS Support Centro de](#) : punto para crear y administrar los casos de AWS Support. También incluye enlaces a otros recursos útiles como foros, preguntas técnicas frecuentes, estado de los servicios y AWS Trusted Advisor.
- [AWS Support](#): página web principal para obtener información sobre AWS Support one-on-one, un canal de soporte de respuesta rápida que lo ayudará a crear y ejecutar aplicaciones en la nube.
- [Contacte con nosotros](#) – Un punto central de contacto para las consultas relacionadas con la facturación AWS, cuentas, eventos, abuso y demás problemas.
- [AWS Términos del sitio de](#) : información detallada sobre nuestros derechos de autor y marca comercial, su cuenta, licencia y acceso al sitio, entre otros temas.

Historial de revisión de la guía del usuario

En la siguiente tabla, se describe la documentación de la documentación de esta versión de AWS Elemental MediaStore. Para obtener notificaciones sobre las actualizaciones de esta documentación, puede suscribirse a una fuente RSS.

Cambio	Descripción	Fecha
Mejora del control de acceso a origen (OAC)	Se trata de información acerca de cómo utilizar OAC con AWS Elemental MediaStore.	17 de abril de 2023
Actualizaciones de cuotas	Se corrigió el valor de la cuota y la descripción de Rules in a Metric Policy.	25 de octubre de 2022
ExpiresAt campo	Los registros de acceso ahora incluyen un ExpiresAt campo que indica la fecha y la hora de caducidad del objeto en función de las reglas de datos transitorios de la política de ciclo de vida del contenido.	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 establezca 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

<u>Contenedor vacío</u>	Ahora puede eliminar todos los objetos dentro de un contenedor a la vez.	7 de abril de 2020
<u>Support para CloudWatch métricas de Amazon</u>	Puede establecer una política de métricas para determinar a qué métricas se MediaStore envían CloudWatch.	30 de marzo de 2020
<u>Comodín en las reglas de eliminación de objetos</u>	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
<u>Políticas del ciclo de vida de los objetos</u>	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

[Compatibilidad con AWS CloudFormation](#)

A partir de ahora, puede utilizar una plantilla de AWS CloudFormation para crear un contenedor de forma automática. 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 subida de contenido en 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	MediaStore Ahora, AWS Elemental admite el registro de acceso, que brinda registros detallados para las solicitudes realizadas a los objetos de un contenedor.	25 de febrero de 2019
Políticas del ciclo de vida de los 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
Mayor cuota de tamaño de objeto	La cuota de tamaño de un objeto ahora es de 25 MB.	10 de octubre de 2018
Mayor cuota de tamaño de objeto	La cuota de tamaño de un objeto ahora es de 20 MB.	6 de septiembre de 2018
Integración de AWS CloudTrail	El contenido CloudTrail de la integración se ha actualizado para alinearlos con los cambios recientes del CloudTrail servicio.	12 de julio de 2018
Colaboración con 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](#)

MediaStore Ahora, AWS Elemental admite Compartir recursos entre orígenes (CORS), lo que permite que las aplicaciones web de los clientes cargadas 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 ServiciosAWS multimedia no están diseñados ni destinados a utilizarse con aplicaciones ni en situaciones que requieran un rendimiento a prueba de fallos, como operaciones de seguridad personal, sistemas de navegación o comunicación, control de tráfico aéreo o 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 materiales o ambientales.

Glosario de AWS

Para ver la terminología más reciente de AWS, consulte el [Glosario de AWS](#) en la Referencia de Glosario de AWS.

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