



Guia do usuário

AWS Elemental MediaStore



AWS Elemental MediaStore: Guia do usuário

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

.....	vi
O que MediaStore é	1
Conceitos e terminologia	1
Serviços relacionados	3
Acessando MediaStore	3
Preços	4
Regiões e endpoints	4
Configuração do AWS Elemental MediaStore	5
Inscreva-se para um Conta da AWS	5
Criar um usuário com acesso administrativo	6
Conceitos básicos	8
Etapa 1: Acesse o AWS Elemental MediaStore	8
Etapa 2: Criar um contêiner	8
Etapa 3: Fazer upload de um objeto	9
Etapa 4: Acessar um objeto	10
Contêineres	11
Regras para nomes de contêineres	11
Criar um contêiner	11
Visualizar detalhes do contêiner	13
Visualizar uma lista de contêineres	14
Excluir um contêiner	15
Políticas	16
Políticas de contêiner	16
Visualizar uma política de contêiner	17
Editar uma política de contêiner	18
Exemplos de políticas de contêiner	19
Políticas de CORS	25
Cenários de casos de uso	26
Adicionar uma política de CORS	27
Visualizar uma política de CORS	28
Editar uma política de CORS	29
Excluir uma política de CORS	30
Solução de problemas	31
Exemplos de políticas de CORS	32

Políticas de ciclo de vida de objetos	33
Componentes de uma política de ciclo de vida de objetos	34
Adicionar uma política de ciclo de vida de objetos	41
Visualizar uma política de ciclo de vida de objetos	43
Editar uma política de ciclo de vida de objetos	44
Excluir uma política de ciclo de vida de objetos	45
Exemplos de política de ciclo de vida de objetos	45
Políticas de métrica	50
Adicionar uma política de métrica	51
Visualizar uma política de métrica	51
Editar uma política de métrica	51
Exemplos de políticas de métrica	52
Pastas	56
Regras para nomes de pastas	56
Criar uma pasta	57
Excluir uma pasta	57
Objetos	58
Fazer upload de um objeto	58
Visualizar uma lista	60
Visualizar detalhes do objeto	63
Fazer download de um objeto	64
Excluir objetos	65
Excluir um objeto	65
Esvaziar um contêiner	66
Segurança	68
Proteção de dados	69
Criptografia de dados	70
Gerenciamento de Identidade e Acesso	70
Público	71
Autenticação com identidades	71
Gerenciar o acesso usando políticas	75
Como o AWS Elemental MediaStore funciona com o IAM	78
Exemplos de políticas baseadas em identidade	85
Solução de problemas	88
Registro em log e monitoramento	90
CloudWatch Alarmes da Amazon	90

AWS CloudTrail troncos	91
AWS Trusted Advisor	91
Validação de conformidade	91
Resiliência	92
Segurança da infraestrutura	93
Prevenção contra o ataque do “substituto confuso” em todos os serviços	93
Monitorar e atribuir tags (tagging)	96
Registrar chamadas de API com o CloudTrail	97
MediaStoreInformações em CloudTrail	97
Exemplo: entradas de arquivo de log	99
Monitoramento com CloudWatch	100
CloudWatch Registros	101
CloudWatch Eventos	111
CloudWatch métricas	114
Tags	119
Recursos compatíveis no AWS Elemental MediaStore	120
Convenções de uso e nomenclatura de tags	120
Como gerenciar tags	120
Trabalhando com CDNs	122
Permitir que o CloudFront acesse seu contêiner	122
Uso do controle de acesso de origem (OAC)	123
Uso de segredos compartilhados	123
Interação do MediaStore com caches HTTP	126
Solicitações condicionais	126
Trabalhando com AWS SDKs	128
Exemplos de código	130
Conceitos básicos	130
Ações	131
Cotas	153
Informações relacionadas	156
Histórico de documento	157
AWS Glossário	162

Aviso de fim do suporte: em 13 de novembro de 2025, o suporte para o AWS MediaStore Elemental AWS será interrompido. Depois de 13 de novembro de 2025, você não poderá mais acessar o MediaStore console ou MediaStore os recursos. Veja esta [postagem em blog](#) para obter mais informações.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.

O que é o AWS Elemental MediaStore?

O AWS Elemental MediaStore é um serviço de originação e armazenamento de vídeo que oferece o alto desempenho e a consistência imediata necessários para a originação ao vivo. Com MediaStore, você pode gerenciar ativos de vídeo como objetos em contêineres para criar fluxos de trabalho de mídia confiáveis e baseados na nuvem.

Para usar o serviço, faça upload de seus objetos a partir de uma origem, como um codificador ou feed de dados, para um contêiner criado no MediaStore.

MediaStore é uma ótima opção para armazenar arquivos de vídeo fragmentados quando você precisa de consistência forte, leituras e gravações de baixa latência e a capacidade de lidar com grandes volumes de solicitações simultâneas. Se você não entrega vídeos de streaming ao vivo, considere usar o [Amazon Simple Storage Service \(Amazon S3\)](#) em vez disso.

Tópicos

- [MediaStore Conceitos e terminologia do AWS Elemental](#)
- [Serviços relacionados](#)
- [Acessando o AWS Elemental MediaStore](#)
- [Preços do AWS Elemental MediaStore](#)
- [Regiões e endpoints para o AWS Elemental MediaStore](#)

MediaStore Conceitos e terminologia do AWS Elemental

ARN

Um [Nome de recurso da Amazon](#).

Corpo

Os dados a serem carregados em um objeto.

Intervalo (bytes)

Um subconjunto de dados do objeto a ser endereçado. Para obter mais informações, consulte [intervalo](#) da especificação HTTP.

Contêiner

Um namespace que contém objetos. Um contêiner possui um endpoint que você pode usar para gravar e recuperar objetos, além de anexar políticas de acesso.

Endpoint

Um ponto de entrada para o MediaStore serviço, fornecido como uma URL raiz HTTPS.

ETag

Uma [tag de entidade](#), que é um hash de dados de objeto.

Pasta

Uma divisão de um contêiner. Uma pasta pode conter objetos e outras pastas.

Item

Um termo usado para se referir a objetos e pastas.

Objeto

Um ativo, semelhante a um [objeto do Amazon S3](#). Os objetos são entidades fundamentais armazenadas no MediaStore. O serviço aceita todos os tipos de arquivo.

Serviço de origem

MediaStore é considerado um serviço de origem porque é o ponto de distribuição para entrega de conteúdo de mídia.

Path

Um identificador exclusivo para um objeto ou pasta, que indica sua localização no contêiner.

Peça

Um subconjunto de dados (bloco) de um objeto.

Política

Uma [política do IAM](#).

Recurso

Uma entidade na AWS com a qual você pode trabalhar. Cada recurso da AWS recebe um nome de recurso da Amazon (ARN) que atua como um identificador exclusivo. Em MediaStore, esse é o recurso e seu formato ARN:

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

Serviços relacionados

- CloudFrontA Amazon é um serviço global de rede de entrega de conteúdo (CDN) que entrega dados e vídeos com segurança aos seus espectadores. Use o CloudFront para entregar conteúdo com o melhor desempenho possível. Para obter mais informações, consulte o [Amazon CloudFront Developer Guide](#).
- AWS CloudFormationé um serviço que ajuda você a modelar e configurar seus AWS recursos. Você cria um modelo que descreve todos os AWS recursos que deseja (como MediaStore contêineres) e AWS CloudFormation cuida do provisionamento e da configuração desses recursos para você. Você não precisa criar e configurar AWS recursos individualmente e descobrir o que depende do quê; AWS CloudFormation lida com tudo isso. Para obter mais informações, consulte o [Guia do usuário do AWS CloudFormation](#).
- AWS CloudTrailé um serviço que permite monitorar as chamadas feitas para a CloudTrail API da sua conta, incluindo chamadas feitas pelo AWS Management Console e outros serviços. AWS CLI Para obter mais informações, consulte o [Guia do usuário do AWS CloudTrail](#).
- A Amazon CloudWatch é um serviço de monitoramento dos recursos da AWS nuvem e dos aplicativos nos quais você executa AWS. Use CloudWatch Eventos para rastrear mudanças no status de contêineres e objetos em MediaStore. Para obter mais informações, consulte a [CloudWatch documentação da Amazon](#).
- AWS Identity and Access Management (IAM) é um serviço web que ajuda você a controlar com segurança o acesso aos AWS recursos para seus usuários. Use o IAM para controlar quem pode usar seus AWS recursos (autenticação) e quais recursos os usuários podem usar de quais formas (autorização). Para obter mais informações, consulte [Configuração do AWS Elemental MediaStore](#).
- O Amazon Simple Storage Service (Amazon S3) é um armazenamento de objeto feito para armazenar e recuperar qualquer quantidade de dados de qualquer lugar. Para mais informações, consulte a [documentação do Amazon S3](#).

Acessando o AWS Elemental MediaStore

Você pode acessar MediaStore usando qualquer um dos seguintes métodos:

- Console de Gerenciamento da AWS — Os procedimentos deste guia explicam como usar o Console de Gerenciamento da AWS para realizar tarefas para MediaStore. Para acessar MediaStore usando o console:

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

- AWS Command Line Interface – Para obter mais informações, consulte o [Guia do usuário do AWS Command Line Interface](#). Para acessar MediaStore usando o endpoint da CLI:

```
aws mediastore
```

- MediaStore API — Se você estiver usando uma linguagem de programação para a qual um SDK não está disponível, consulte a [Referência da AWS Elemental MediaStore API](#) para obter informações sobre as ações da API e sobre como fazer solicitações de API. Para acessar MediaStore usando o endpoint da API REST:

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

- AWS SDKs — Se você estiver usando uma linguagem de programação para a qual a AWS fornece um SDK, você pode usar um SDK para acessar. MediaStore SDKs simplifique a autenticação, integre-se facilmente ao seu ambiente de desenvolvimento e forneça acesso fácil aos MediaStore comandos. Para obter mais informações, consulte [Ferramentas para a Amazon Web Services](#).
- AWS Tools for Windows PowerShell — Para obter mais informações, consulte o [Guia Ferramentas da AWS para PowerShell do usuário](#).

Preços do AWS Elemental MediaStore

Assim como em outros AWS produtos, não há contratos ou compromissos mínimos de uso MediaStore. É cobrada uma taxa por GB consumidos quando o conteúdo entra no serviço e uma taxa mensal por GB para o conteúdo armazenado no serviço. Para obter mais informações, consulte os preços do [AWS Elemental MediaStore](#).

Regiões e endpoints para o AWS Elemental MediaStore

Para reduzir a latência de dados em seus aplicativos, MediaStore oferece um endpoint regional para fazer sua solicitação:

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

Para ver a lista completa das regiões da AWS onde MediaStore está disponível, consulte os [MediaStore endpoints e cotas do AWS Elemental na Referência](#) geral da AWS.

Configuração do AWS Elemental MediaStore

Esta seção orienta você pelas etapas necessárias para configurar os usuários para acessar o AWS Elemental MediaStore. Para obter informações básicas e adicionais sobre gerenciamento de identidade e acesso para MediaStore, consulte [Identity and Access Management para AWS Elemental MediaStore](#).

Para começar a usar o AWS Elemental MediaStore, conclua as etapas a seguir.

Tópicos

- [Inscreva-se para um Conta da AWS](#)
- [Criar um usuário com acesso administrativo](#)

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a <https://portal.aws.amazon.com/billing/inscrição>.
2. Siga as instruções online.

Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar [tarefas que exigem acesso de usuário-raiz](#).

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando <https://aws.amazon.com/e> escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login [AWS Management Console](#) como proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

Para obter ajuda ao fazer login usando o usuário-raiz, consulte [Fazer login como usuário-raiz](#) no Guia do usuário do Início de Sessão da AWS .

2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

Para obter instruções, consulte [Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz \(console\) no Guia](#) do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

Para obter instruções, consulte [Habilitar o AWS IAM Identity Center](#) no Guia do usuário do AWS IAM Identity Center .

2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte [Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM](#) no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte [Como fazer login no portal de AWS acesso](#) no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

Para obter instruções, consulte [Criar um conjunto de permissões](#) no Guia do usuário do AWS IAM Identity Center .

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

Para obter instruções, consulte [Adicionar grupos](#) no Guia do usuário do AWS IAM Identity Center .

Comece a usar o AWS Elemental MediaStore

Este tutorial de conceitos básicos mostra como usar o AWS Elemental MediaStore para criar um contêiner e fazer o upload de um objeto.

Tópicos

- [Etapa 1: Acesse o AWS Elemental MediaStore](#)
- [Etapa 2: Criar um contêiner](#)
- [Etapa 3: Fazer upload de um objeto](#)
- [Etapa 4: Acessar um objeto](#)

Etapa 1: Acesse o AWS Elemental MediaStore

Depois de configurar sua conta da AWS e criar usuários e funções, você faz login no console do AWS Elemental MediaStore.

Para acessar o AWS Elemental MediaStore

- Faça login no AWS Management Console e abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.

Note

Você pode fazer login usando qualquer uma das credenciais do IAM que você criou para essa conta. Para obter informações sobre como criar credenciais do IAM, consulte [Configuração do AWS Elemental MediaStore](#).

Etapa 2: Criar um contêiner

Você usa contêineres no AWS Elemental MediaStore para armazenar suas pastas e objetos. Você pode usar contêineres para agrupar objetos relacionados, da mesma forma que usa um diretório para agrupar arquivos em um sistema de arquivos. Você não é cobrado quando cria contêineres, apenas quando faz upload de um objeto para um contêiner.

Para criar um contêiner

1. Na página Containers (Contêineres), selecione Create container (Criar contêiner).
2. Em Container name (nome do contêiner), digite um nome para o contêiner. Para obter mais informações, consulte [Regras para nomes de contêineres](#).
3. Escolha Criar contêiner. O AWS Elemental MediaStore adiciona o novo contêiner a uma lista de contêineres. Inicialmente, o status do contêiner é Creating (Criando) e, depois, ele muda para Active (Ativo).

Etapa 3: Fazer upload de um objeto

Você pode fazer upload de objetos (até 25 MB cada) para um contêiner ou para uma pasta em um contêiner. Para fazer upload de um objeto para uma pasta, você especifica o caminho para a pasta. Se a pasta já existir, o AWS Elemental MediaStore armazena o objeto na pasta. Se a pasta não existir, o serviço a cria e, em seguida, armazena o objeto na pasta.

Note

Os nomes de arquivo de objeto podem conter apenas letras, números, pontos (.), sublinhados (_), tils (~) e hifens (-).

Como fazer upload de um objeto

1. Na página Containers (Contêineres), selecione o nome do contêiner que você acabou de criar. A página de detalhes do contêiner é exibida.
2. Selecione Upload object (Fazer upload de objeto).
3. Em Target path (Caminho de destino), digite o caminho das pastas. Por exemplo, .premium/canada Se alguma das pastas no caminho ainda não existir, o AWS Elemental as MediaStore criará automaticamente.
4. Em Object (Objeto), selecione Browse (Navegar).
5. Navegue até a pasta apropriada e escolha um objeto para fazer upload.
6. Selecione Open (Abrir) e Upload (Fazer upload).

Etapa 4: Acessar um objeto

Você pode fazer download de seus objetos para um endpoint especificado.

1. Na página Containers (Contêineres), selecione o nome do contêiner que tem o objeto do qual você deseja fazer download.
2. Se o objeto do qual você deseja fazer download estiver em uma subpasta, continue escolhendo os nomes das pastas até ver o objeto.
3. Escolha o nome do objeto.
4. Na página de detalhes do objeto, selecione Download (Fazer download).

Contêineres no AWS Elemental MediaStore

Você usa contêineres MediaStore para armazenar suas pastas e objetos. Objetos relacionados podem ser agrupados em contêineres da mesma forma que você usa um diretório para agrupar arquivos em um sistema de arquivos. Você não é cobrado quando cria contêineres, apenas quando faz upload de um objeto para um contêiner. Para obter mais informações sobre cobranças, consulte os preços do [AWS Elemental MediaStore](#).

Tópicos

- [Regras para nomes de contêineres](#)
- [Criar um contêiner](#)
- [Visualizar os detalhes de um contêiner](#)
- [Visualizar uma lista de contêineres](#)
- [Excluir um contêiner](#)

Regras para nomes de contêineres

Ao escolher um nome para o contêiner, lembre-se do seguinte:

- O nome deve ser exclusivo dentro da conta atual para a região da AWS atual.
- O nome pode conter letras maiúsculas, minúsculas, números e sublinhados (_).
- O nome deve ter de 1 a 255 caracteres.
- Os nomes diferenciam letras maiúsculas de minúsculas. Por exemplo, você pode ter um contêiner chamado `myContainer` e uma pasta chamada `mycontainer` porque esses nomes são exclusivos.
- Um contêiner não pode ser renomeados depois de criado.

Criar um contêiner

Você pode criar até 100 contêineres para cada conta da AWS. É possível criar quantas pastas quiser, desde que não estejam aninhadas mais de 10 níveis em um contêiner. Além disso, você pode fazer upload de quantos objetos quiser para cada contêiner.

Tip

Você também pode criar um contêiner automaticamente usando um AWS CloudFormation modelo. O modelo do AWS CloudFormation gerencia os dados para cinco ações de API: criação de um contêiner, definição do registro de acesso, atualização da política de contêiner padrão, adição de uma política de CORS (compartilhamento de recursos entre origens) e adição de uma política de ciclo de vida de objetos. Para obter mais informações, consulte o [Guia do usuário do AWS CloudFormation](#).

Para criar um contêiner (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), selecione Create container (Criar contêiner).
3. Em nome do Container (Contêiner), insira um nome para o contêiner. Para obter mais informações, consulte [Regras para nomes de contêineres](#).
4. Escolha Criar contêiner. O AWS Elemental MediaStore adiciona o novo contêiner a uma lista de contêineres. Inicialmente, o status do contêiner é Creating (Criando) e, depois, ele muda para Active (Ativo).

Para criar um contêiner (AWS CLI)

- No AWS CLI, use o `create-container` comando:

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

O exemplo a seguir mostra o valor de retorno:

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

```
}
```

Visualizar os detalhes de um contêiner

Os detalhes de um contêiner incluem a política, o endpoint, o ARN e a data de criação do contêiner.

Para visualizar os detalhes de um contêiner (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), selecione o nome do contêiner.

A página de detalhes do contêiner é exibida. Esta página é dividida em duas seções:

- A seção Objects (Objetos), que lista os objetos e as pastas no contêiner.
- A seção de política do Container (Contêiner), que mostra a política baseada em recursos associada a esse contêiner. Para obter informações sobre políticas de recursos, consulte [Políticas de contêiner](#).

Para visualizar os detalhes de um contêiner (AWS CLI)

- No AWS CLI, use o `describe-container` comando:

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

O exemplo a seguir mostra o 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"
  }
}
```

```
}
```

Visualizar uma lista de contêineres

Você pode visualizar uma lista de todos os contêineres que estão associados à sua conta.

Para visualizar uma lista de contêineres (console)

- Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.

A página Containers (Contêineres) é exibida, listando todos os contêineres associados à sua conta.

Para visualizar uma lista de contêineres (AWS CLI)

- No AWS CLI, use o `list-containers` comando.

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

O exemplo a seguir mostra o valor de retorno:

```
{
  "Containers": [
    {
      "CreationTime": 1505317931.0,
      "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": 1506528818.0,
      "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"  
    }  
]  
}
```

Excluir um contêiner

Você só pode excluir um contêiner se ele não tiver objetos.

Para excluir um contêiner (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), selecione a opção à esquerda do nome do contêiner.
3. Escolha Excluir.

Para excluir um contêiner (AWS CLI)

- No AWS CLI, use o `delete-container` comando:

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

Este comando não tem valor de retorno.

Políticas no AWS Elemental MediaStore

Você pode aplicar uma ou mais dessas políticas ao seu contêiner do AWS Elemental MediaStore :

- [Política de contêiner](#) - define direitos de acesso a todas as pastas e objetos dentro do contêiner. MediaStore define uma política padrão que permite aos usuários realizar todas as MediaStore operações no contêiner. Essa política especifica que todas as operações devem ser executadas por HTTPS. Depois de criar um contêiner, você poderá editar a política de contêiner.
- [Política de compartilhamento de recursos de origem cruzada \(CORS\)](#) - permite que aplicativos web clientes de um domínio interajam com recursos em um domínio diferente. MediaStore não define uma política CORS padrão.
- [Política de métricas](#) - Permite MediaStore enviar métricas para a Amazon CloudWatch. MediaStore não define uma política métrica padrão.
- [Política de ciclo de vida do objeto](#) - controla por quanto tempo os objetos permanecem em um MediaStore contêiner. MediaStore não define uma política padrão de ciclo de vida do objeto.

Políticas de contêineres no AWS Elemental MediaStore

Cada contêiner tem uma política baseada em recursos que controla os direitos de acesso a todas as pastas e objetos desse contêiner. A política padrão, que é anexada automaticamente a todos os novos contêineres, permite o acesso a todas as MediaStore operações do AWS Elemental no contêiner. Ela especifica que esse acesso tem a condição de exigir HTTPS para as operações. Depois de criar um contêiner, você pode editar a política anexada a ele.

Também é possível especificar uma [política de ciclo de vida de objetos](#) que controla a data de expiração de objetos em um contêiner. Depois de os objetos chegarem à idade máxima especificada, o serviço excluirá os objetos do contêiner.

Tópicos

- [Visualizar uma política de contêiner](#)
- [Editar uma política de contêiner](#)
- [Exemplos de políticas de contêiner](#)

Visualizar uma política de contêiner

Você pode usar o console ou o AWS CLI para visualizar a política baseada em recursos de um contêiner.

Para visualizar uma política de contêiner (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), escolha o nome do contêiner.

A página de detalhes do contêiner é exibida. A política é exibida na seção Container policy (Política de contêiner).

Para visualizar uma política de contêiner (AWS CLI)

- No AWS CLI, use o `get-container-policy` comando:

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

O exemplo a seguir mostra o 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"
          }
        }
      }
    ]
  }
}
```

```
    }
  }
}
]
```

Editar uma política de contêiner

Você pode editar as permissões na política de contêiner padrão ou pode criar uma nova política que substitui a política padrão. Leva até cinco minutos para que a nova política tenha efeito.

Para editar uma política de contêiner (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), escolha o nome do contêiner.
3. Escolha Editar política. Para obter exemplos que mostram como definir permissões diferentes, consulte [the section called “Exemplos de políticas de contêiner”](#).
4. Faça as alterações apropriadas e, em seguida, selecione Save (Salvar).

Para editar uma política de contêiner (AWS CLI)

1. Crie um arquivo que defina a política de contêiner:

```
{
  "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. No AWS CLI, use o `put-container-policy` comando:

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

Este comando não tem valor de retorno.

Exemplos de políticas de contêiner

Os exemplos a seguir mostram políticas de contêiner construídas para diferentes grupos de usuários.

Tópicos

- [Exemplo de política de contêiner: padrão](#)
- [Exemplo de política de contêiner: acesso público de leitura por HTTPS](#)
- [Exemplo de política de contêiner: acesso público de leitura por HTTP ou HTTPS](#)
- [Exemplo de política de contêiner: acesso de leitura entre contas – HTTP habilitado](#)
- [Exemplo de política de contêiner: acesso de leitura entre contas por HTTPS](#)
- [Exemplo de política de contêiner: acesso de leitura entre contas a uma função](#)
- [Exemplo de política de contêiner: acesso total entre contas a uma função](#)
- [Exemplo de política de contêiner: acesso restrito a endereços IP específicos](#)

Exemplo de política de contêiner: padrão

Quando você cria um contêiner, o AWS Elemental anexa MediaStore automaticamente a seguinte política baseada em 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" }
    }
  }
]
}

```

A política é criada no serviço, de modo que você não precisa criá-la. Porém, você pode [editar a política](#) no contêiner se as permissões na política padrão não estiverem alinhadas com as permissões que você deseja usar para o contêiner.

A política padrão que é atribuída a todos os novos contêineres permite acesso a todas as operações do MediaStore no contêiner. Ela especifica que esse acesso tem a condição de exigir HTTPS para as operações.

Exemplo de política de contêiner: acesso público de leitura por HTTPS

Essa política de exemplo permite que os usuários recuperem um objeto por meio de uma solicitação HTTPS. Ela permite acesso de leitura a qualquer pessoa por meio de uma conexão segura SSL/TLS: usuários autenticados e usuários anônimos (usuários que não estiverem conectados). A instrução tem o nome `PublicReadOverHttps`. Permite acesso às operações `GetObject` e `DescribeObject` em qualquer objeto, conforme especificado pelo `*` no final do caminho do recurso. Ela especifica que esse acesso tem a condição de exigir HTTPS para as operações:

```

{
  "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"
        }
      }
    }
  ]
}

```

```

    }
  ]
}

```

Exemplo de política de contêiner: acesso público de leitura por HTTP ou HTTPS

Essa política de exemplo permite acesso às operações `GetObject` e `DescribeObject` em qualquer objeto (conforme especificado pelo `*` no final do caminho do recurso). Ela permite acesso de leitura a qualquer pessoa, incluindo todos os usuários autenticados e usuários anônimos (usuários que não estão conectados):

```

{
  "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"] }
      }
    }
  ]
}

```

Exemplo de política de contêiner: acesso de leitura entre contas – HTTP habilitado

Essa política de exemplo permite que os usuários recuperem um objeto por meio de uma solicitação HTTP. Ela permite esse acesso a usuários autenticados com acesso entre contas. O objeto não precisa estar hospedado em um servidor com um 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" ]
      }
    }
  } ]
}

```

Exemplo de política de contêiner: acesso de leitura entre contas por HTTPS

Este exemplo de política permite acessar as operações `GetObject` e `DescribeObject` em qualquer objeto (conforme especificado pelo `*` no final do caminho do recurso) que pertence ao usuário raiz do <outro número de conta> especificado. Ela especifica que esse acesso tem a condição de exigir HTTPS para as operações:

```

{
  "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"
        }
      }
    }
  ]
}

```

Exemplo de política de contêiner: acesso de leitura entre contas a uma função

O exemplo de política permite acessar as operações `GetObject` e `DescribeObject` em qualquer objeto (conforme especificado pelo `*` no final do caminho do recurso) que pertence ao <número de

conta do proprietário>. Ela permite esse acesso a qualquer usuário do <outro número de conta> se essa conta tiver assumido a função especificada no <nome da função>:

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

Exemplo de política de contêiner: acesso total entre contas a uma função

Essa política de exemplo permite acesso entre contas para atualizar qualquer objeto na conta, desde que o usuário esteja conectado por HTTP. Ela também permite acesso entre contas para excluir, fazer download e descrever os objetos por HTTP ou HTTPS em uma conta que tenha assumido a função especificada:

- A primeira instrução é `CrossAccountRolePostOverHttps`. Ela permite o acesso à operação `PutObject` em qualquer objeto e permite esse acesso a qualquer usuário da conta especificada, se essa conta tiver assumido a função especificada em <nome da função>. Ela especifica que esse acesso tem a condição de exigir HTTPS para a operação, essa condição deve sempre ser incluída ao fornecer acesso à operação `PutObject`.

Em outras palavras, qualquer principal que tenha acesso a contas cruzadas pode acessar a operação `PutObject`, mas somente por HTTPS.

- A segunda instrução é `CrossAccountFullAccessExceptPost`. Permite acesso a todas as operações, exceto `PutObject` em qualquer objeto. Permite esse acesso a qualquer usuário da conta especificada se essa conta tiver assumido a função especificada no <nome da função>. Esse acesso não tem a condição de exigir HTTPS para as operações.

Em outras palavras, qualquer conta que tenha acesso entre contas pode acessar `DeleteObject`, `GetObject` e assim por diante, mas não `PutObject`, e pode fazer isso por HTTP ou HTTPS.

Se você não excluir a operação `PutObject` da segunda instrução, a instrução não será válida, porque se você incluir `PutObject`, é necessário definir explicitamente o HTTPS como uma condição.

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

Exemplo de política de contêiner: acesso restrito a endereços IP específicos

Esse exemplo de política permite acesso a todas as MediaStore operações do AWS Elemental em objetos no contêiner especificado. No entanto, a solicitação deve se originar no intervalo de endereços IP especificados na condição.

A condição nesta declaração identifica o intervalo 198.51.100.* de endereços IP permitidos do Internet Protocol versão 4 (IPv4), com uma exceção: 198.51.100.188.

O bloco da `Condition` usa as condições `IpAddress` e `NotIpAddress` e a chave de condição `aws:SourceIp`, que é uma chave de condição que abrange toda a AWS. Os `aws:sourceIp` IPv4 valores usam a notação CIDR padrão. Para obter mais informações, consulte [Operadores de condição de endereço IP](#) no Guia do usuário do IAM.

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

Políticas de compartilhamento de recursos de origem cruzada (CORS) no AWS Elemental MediaStore

O compartilhamento de recursos de origem cruzada (CORS) define uma maneira de os aplicativos web clientes carregados em um domínio interagirem com recursos em outro domínio. Com o suporte do CORS no AWS MediaStore Elemental, você pode criar aplicativos web avançados do lado

do cliente e permitir seletivamente o acesso de MediaStore origem cruzada aos seus recursos.
MediaStore

Note

Se você estiver usando CloudFront a Amazon para distribuir conteúdo de um contêiner que tenha uma política de CORS, certifique-se de [configurar a distribuição para o AWS MediaStore Elemental](#) (incluindo a etapa de editar o comportamento do cache para configurar o CORS).

Esta seção fornece uma visão geral do CORS. Os subtópicos descrevem como você pode habilitar o CORS usando o console AWS MediaStore Elemental ou programaticamente usando a API MediaStore REST e a AWS. SDKs

Tópicos

- [Cenários de casos de uso do CORS](#)
- [Adicionar uma política de CORS a um contêiner](#)
- [Visualizar uma política de CORS](#)
- [Editar uma política de CORS](#)
- [Excluir uma política de CORS](#)
- [Solução de problemas de CORS](#)
- [Exemplos de políticas de CORS](#)

Cenários de casos de uso do CORS

Veja a seguir exemplos de cenário de uso do CORS:

- Cenário 1: suponha que você esteja distribuindo vídeo de streaming ao vivo em um contêiner do AWS MediaStore Elemental chamado. LiveVideo Seus usuários carregam o endpoint de manifesto do vídeo `http://livevideo.mediastore.ap-southeast-2.amazonaws.com` de uma origem específica, como `www.example.com`. Você quer usar um reprodutor de JavaScript vídeo para acessar vídeos originados desse contêiner por meio de solicitações não autenticadas GET. PUT Normalmente, um navegador bloqueia JavaScript a permissão dessas solicitações, mas você pode definir uma política de CORS em seu contêiner para habilitar explicitamente essas solicitações. `www.example.com`

- Cenário 2: suponha que você queira hospedar a mesma transmissão ao vivo do Cenário 1 do seu MediaStore contêiner, mas queira permitir solicitações de qualquer origem. Você pode configurar uma política de CORS para permitir origens curinga (*), para que solicitações de qualquer origem possam acessar o vídeo.

Adicionar uma política de CORS a um contêiner

Esta seção explica como adicionar uma configuração de compartilhamento de recursos de origem cruzada (CORS) a um contêiner do AWS MediaStore Elemental. O CORS permite que aplicativos web do cliente que sejam carregados em um domínio interajam em outro domínio.

Para configurar seu contêiner para permitir solicitações de origem cruzada, adicione uma política de CORS ao contêiner. Uma política de CORS define regras que identificam as origens que você permite que acessem seu contêiner, as operações (métodos HTTP) compatíveis para cada origem e outras informações específicas da operação.

Quando você adiciona uma política de CORS ao contêiner, as [políticas de contêiner](#) (que controlam os direitos de acesso ao contêiner), continuam a ser aplicadas.

Para adicionar uma política de CORS (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), selecione o nome do contêiner para o qual você deseja criar uma política de CORS.

A página de detalhes do contêiner é exibida.

3. Na seção Container CORS policy (Política de CORS de contêiner), selecione Create CORS policy (Criar política de CORS).
4. Insira a política no formato JSON e, em seguida, selecione Save (Salvar).

Para adicionar uma política de CORS (AWS CLI)

1. Crie um arquivo que defina a política de CORS:

```
[
  {
    "AllowedHeaders": [
      "*"
    ]
  }
]
```

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

2. No AWS CLI, use o `put-cors-policy` comando.

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

Este comando não tem valor de retorno.

Visualizar uma política de CORS

O compartilhamento de recursos de origem cruzada (CORS) define uma maneira de os aplicativos web clientes carregados em um domínio interagirem com recursos em outro domínio.

Para visualizar uma política de CORS (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), selecione o nome do contêiner para o qual você deseja visualizar a política de CORS.

A página de detalhes do contêiner será exibida, com a política de CORS na seção Container CORS policy (Política de CORS de contêiner).

Para visualizar uma política de CORS (AWS CLI)

- No AWS CLI, use o `get-cors-policy` comando:

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

O exemplo a seguir mostra o valor de retorno:

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

Editar uma política de CORS

O compartilhamento de recursos de origem cruzada (CORS) define uma maneira de os aplicativos web clientes carregados em um domínio interagirem com recursos em outro domínio.

Para editar uma política de CORS (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), selecione o nome do contêiner para o qual você deseja editar a política de CORS.

A página de detalhes do contêiner é exibida.

3. Na seção Container CORS policy (Política de CORS de contêiner), selecione Edit CORS policy (Editar política de CORS).
4. Faça as alterações na política e selecione Save (Salvar).

Para editar uma política de CORS (AWS CLI)

1. Crie um arquivo que defina a política CORS atualizada:

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

2. No AWS CLI, use o `put-cors-policy` comando.

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

Este comando não tem valor de retorno.

Excluir uma política de CORS

O compartilhamento de recursos de origem cruzada (CORS) define uma maneira de os aplicativos web clientes carregados em um domínio interagirem com recursos em outro domínio. Excluir a política de CORS de um contêiner remove permissões para solicitações de origem cruzada.

Para excluir uma política de CORS (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), selecione o nome do contêiner para o qual você deseja excluir a política de CORS.

A página de detalhes do contêiner é exibida.

3. Na seção Container CORS policy (Política de CORS de contêiner), selecione Delete CORS policy (Excluir política de CORS).
4. Escolha Continue (Continuar) para confirmar e escolha Save (Salvar).

Para excluir uma política de CORS (AWS CLI)

- No AWS CLI, use o `delete-cors-policy` comando:

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

Este comando não tem valor de retorno.

Solução de problemas de CORS

Se você notar um comportamento inesperado ao acessar um contêiner que tem uma política de CORS, siga estas etapas para solucionar o problema.

1. Verifique se a política de CORS está anexada ao contêiner.

Para obter instruções, consulte [the section called “Visualizar uma política de CORS”](#).

2. Capture a solicitação e a resposta completas usando uma ferramenta de sua escolha, como o console do desenvolvedor do navegador. Verifique se a política de CORS anexada ao contêiner inclui pelo menos uma regra de CORS que corresponda aos dados na solicitação, da seguinte maneira:
 - a. Verifique se a solicitação tem um cabeçalho `Origin`.

Se o cabeçalho estiver ausente, o AWS Elemental MediaStore não tratará a solicitação como uma solicitação de origem cruzada e não enviará os cabeçalhos de resposta do CORS de volta na resposta.

- b. Verifique se o cabeçalho `Origin` na solicitação corresponde a pelo menos um dos elementos `AllowedOrigins` na `CORSRule` específica.

O esquema, o host e os valores de porta no cabeçalho da solicitação `Origin` devem corresponder a `AllowedOrigins` na `CORSRule`. Por exemplo, se você definir a `CORSRule` para permitir a origem `http://www.example.com`, as origens `https://`

`www.example.com` e `http://www.example.com:80` da solicitação não corresponderão à origem permitida na configuração.

- c. Verifique se o método na solicitação (ou o método especificado em `Access-Control-Request-Method` no caso de uma solicitação de simulação) é um dos elementos `AllowedMethods` na mesma `CORSRule`.
- d. Para uma solicitação de simulação, se a solicitação incluir um cabeçalho `Access-Control-Request-Headers`, verifique se `CORSRule` inclui as entradas `AllowedHeaders` para cada valor no cabeçalho `Access-Control-Request-Headers` header.

Exemplos de políticas de CORS

Os exemplos a seguir mostram as políticas de compartilhamento de recursos de origem cruzada (CORS).

Tópicos

- [Exemplo de política de CORS: acesso de leitura para qualquer domínio](#)
- [Exemplo de política de CORS: acesso de leitura para um domínio específico](#)

Exemplo de política de CORS: acesso de leitura para qualquer domínio

A política a seguir permite que uma página da web de qualquer domínio recupere conteúdo do seu contêiner do AWS MediaStore Elemental. A solicitação inclui todos os cabeçalhos HTTP do domínio de origem, e o serviço responde apenas às solicitações HTTP GET e HTTP HEAD do domínio de origem. Os resultados são armazenados em cache por 3.000 segundos antes de um novo conjunto de resultados ser fornecido.

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

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

Exemplo de política de CORS: acesso de leitura para um domínio específico

A política a seguir permite que uma página da `https://www.example.com` web recupere conteúdo do seu contêiner do AWS MediaStore Elemental. A solicitação inclui todos os cabeçalhos HTTP do `https://www.example.com`, e o serviço responde apenas às solicitações HTTP GET e HTTP HEAD do `https://www.example.com`. Os resultados são armazenados em cache por 3.000 segundos antes de um novo conjunto de resultados ser fornecido.

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

Políticas de ciclo de vida de objetos no AWS Elemental MediaStore

Para cada contêiner, você pode criar uma política de ciclo de vida de objetos que controla por quanto tempo os objetos devem ser armazenados no contêiner. Quando os objetos atingem a idade máxima especificada, o AWS Elemental MediaStore exclui os objetos. Você pode excluir objetos quando eles não forem mais necessários para economizar custos de armazenamento.

Você também pode especificar que os objetos MediaStore devem ser movidos para a classe de armazenamento de acesso infrequente (IA) após atingirem uma certa idade. Os objetos armazenados na classe de armazenamento IA têm taxas de armazenamento e recuperação

diferentes dos objetos armazenados na classe de armazenamento padrão. Para obter mais informações, consulte [Preços do MediaStore](#).

Uma política de ciclo de vida de objetos contém regras, que determinam o ciclo de vida de objetos por subpasta. (Você não pode atribuir uma política de ciclo de vida de objetos a objetos individuais). Você pode anexar apenas uma política de ciclo de vida de objetos a um contêiner, mas você pode adicionar até 10 regras a cada política de ciclo de vida de objeto. Para obter mais informações, consulte [Componentes de uma política de ciclo de vida de objetos](#).

Tópicos

- [Componentes de uma política de ciclo de vida de objetos](#)
- [Adicionar uma política de ciclo de vida de objetos a um contêiner](#)
- [Visualizar uma política de ciclo de vida de objetos](#)
- [Editar uma política de ciclo de vida de objetos](#)
- [Excluir uma política de ciclo de vida de objetos](#)
- [Exemplos de política de ciclo de vida de objetos](#)

Componentes de uma política de ciclo de vida de objetos

As políticas de ciclo de vida dos objetos governam por quanto tempo os objetos permanecem em um contêiner do AWS Elemental MediaStore. Cada política de ciclo de vida de objetos consiste em uma ou mais regras, que determinam o ciclo de vida de objetos. Uma regra pode ser aplicada a uma pasta, várias pastas ou a todo o contêiner.

Você pode anexar uma política de ciclo de vida de objeto a um contêiner, e cada política de ciclo de vida de objetos pode conter até 10 regras. Você não pode atribuir uma política de ciclo de vida de objetos a um único objeto.

Regras em uma política de ciclo de vida de objetos

É possível criar três tipos de regras:

- [Dados temporários](#)
- [Exclusão de objeto](#)
- [Transição do ciclo de vida](#)

Dados temporários

Uma regra de dados temporários define objetos para expirarem em segundos. Esse tipo de regra se aplica somente a objetos adicionados ao contêiner após a diretiva entrar em vigor. São necessários até 20 minutos MediaStore para aplicar a nova política ao contêiner.

Um exemplo de uma regra para dados temporários é semelhante ao seguinte:

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

As regras de dados temporários têm três partes:

- `path`: sempre definido como `wildcard`. Use essa parte para definir quais objetos deseja excluir. É possível usar um ou mais curingas, representados por um asterisco (*). Cada curinga representa qualquer combinação de zero ou mais caracteres. Por exemplo, `"path": [{"wildcard": "Football/index*.m3u8"}]`, aplica-se a todos os arquivos na pasta `Football` que correspondem ao padrão de `index*.m3u8` (como `index.m3u8`, `index1.m3u8` e `index123456.m3u8`). É possível incluir até 10 caminhos em uma única regra.
- `seconds_since_create`: sempre definido como `numeric`. É possível especificar um valor de 1 a 300 segundos. Também é possível definir o operador como “maior que” (>) ou “maior que ou igual a” (>=).
- `action`: sempre definido como `EXPIRE`.

Para regras de dados temporários (objetos expiram em segundos), não há atraso entre a expiração de um objeto e a exclusão do objeto.

Note

Os objetos que estão sujeitos a uma regra de dados temporários não são incluídos em uma resposta `list-items`. Além disso, objetos que expiram devido a uma regra de dados transitória não emitem um CloudWatch evento quando expiram.

Exclusão de objeto

Uma regra de exclusão de objeto define objetos para expirar em dias. Esse tipo de regra se aplica a todos os objetos do contêiner, mesmo que tenham sido adicionados a ele antes da criação da política. A aplicação da nova política leva até 20 minutos, mas pode levar até 24 horas para que os objetos sejam retirados do contêiner. MediaStore

Um exemplo de duas regras para excluir objetos é semelhante ao seguinte:

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

As regras de exclusão de objetos têm três partes:

- `path`: defina como `prefix` ou `wildcard`. Você não pode misturar `prefix` e `wildcard` na mesma regra. Se quiser usar ambos, será necessário criar uma regra para `prefix` e uma regra separada para `wildcard`, como mostrado no exemplo acima.

- `prefix` – defina o caminho para `prefix` se quiser excluir todos os objetos de uma pasta específica. Se o parâmetro estiver vazio (`"path": [{ "prefix": "" }],`), o destino será todos os objetos armazenados em qualquer lugar no contêiner atual. É possível incluir até 10 caminhos `prefix` em uma única regra.
- `wildcard` – defina o caminho para `wildcard` se quiser excluir objetos específicos com base no nome de arquivo e/ou tipo de arquivo. É possível usar um ou mais curingas, representados por um asterisco (*). Cada curinga representa qualquer combinação de zero ou mais caracteres. Por exemplo, `"path": [{"wildcard": "Football/*.ts"}]`, aplica-se a todos os arquivos na pasta `Football` que correspondem ao padrão de `*.ts` (como `filename.ts`, `filename1.ts` e `filename123456.ts`). É possível incluir até 10 caminhos `wildcard` em uma única regra.
- `days_since_create`: sempre definido como `numeric`. É possível especificar um valor de 1 a 36.500 dias. Também é possível definir o operador como “maior que” (`>`) ou “maior que ou igual a” (`>=`).
- `action`: sempre definido como `EXPIRE`.

Para excluir regras de objetos (objetos expiram em dias), pode haver um pequeno atraso entre a expiração de um objeto e a exclusão do objeto. No entanto, as alterações no faturamento ocorrem assim que o objeto expira. Por exemplo, se uma regra de ciclo de vida especificar `10 days_since_create`, a conta não será cobrada pelo objeto depois que o objeto tiver 10 dias, mesmo que ele ainda não tenha sido excluído.

Transição do ciclo de vida

Uma regra de transição do ciclo de vida define os objetos a serem movidos para a classe de armazenamento de acesso infrequente (IA) depois que eles atingem uma determinada idade, medida em dias. Os objetos armazenados na classe de armazenamento IA têm taxas de armazenamento e recuperação diferentes dos objetos armazenados na classe de armazenamento padrão. Para obter mais informações, consulte [Preços do MediaStore](#).

Depois que um objeto for movido para a classe de armazenamento IA, não será possível movê-lo de volta para a classe de armazenamento padrão.

A regra de transição do ciclo de vida se aplica a todos os objetos do contêiner, mesmo que tenham sido adicionados a ele antes da criação da política. A aplicação da nova política leva até 20 minutos, mas pode levar até 24 horas para que os objetos sejam retirados do contêiner. MediaStore

Veja a seguir um exemplo de regra de transição do ciclo de vida:

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

As regras de transição do ciclo de vida têm três partes:

- **path**: defina como **prefix** ou **wildcard**. Você não pode misturar **prefix** e **wildcard** na mesma regra. Se você quiser usar ambos, deverá criar uma regra para **prefix** e uma regra separada para **wildcard**.
- **prefix** – defina o caminho como **prefix** se desejar fazer a transição de todos os objetos em uma pasta específica para a classe de armazenamento IA. Se o parâmetro estiver vazio (`"path": [{ "prefix": "" }],`), o destino será todos os objetos armazenados em qualquer lugar no contêiner atual. É possível incluir até 10 caminhos **prefix** em uma única regra.
- **wildcard** – defina o caminho como **wildcard** se desejar fazer a transição de objetos específicos para a classe de armazenamento IA com base no nome do arquivo e/ou no tipo de arquivo. É possível usar um ou mais curingas, representados por um asterisco (*). Cada curinga representa qualquer combinação de zero ou mais caracteres. Por exemplo, `"path": [{"wildcard": "Football/*.ts"}],` aplica-se a todos os arquivos na pasta `Football` que correspondem ao padrão de `*.ts` (como `filename.ts`, `filename1.ts` e `filename123456.ts`). É possível incluir até 10 caminhos **wildcard** em uma única regra.
- **days_since_create**: sempre definido como `"numeric": [">=" , 30]`.
- **action**: sempre definido como `ARCHIVE`.

Exemplo

Vamos supor que um contêiner chamado `LiveEvents` tenha quatro subpastas: `Football`, `Baseball`, `Basketball` e `AwardsShow`. A política de ciclo de vida de objetos atribuída à pasta `LiveEvents` pode ter a seguinte aparência:

```

{
  "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"
}
]
}

```

A política anterior especifica o seguinte:

- A primeira regra instrui o AWS MediaStore Elemental a excluir objetos armazenados na pasta e LiveEvents/Football LiveEvents/Baseball na pasta após terem mais de 28 dias.
- A segunda regra instrui o serviço a excluir objetos armazenados na pasta LiveEvents/AwardsShow ao completarem 15 dias ou mais.
- A terceira regra instrui o serviço a excluir objetos armazenados em qualquer lugar no contêiner LiveEvents depois de completarem 40 dias. Essa regra se aplica a objetos armazenados diretamente no contêiner LiveEvents, bem como objetos armazenados em qualquer uma das quatro subpastas do contêiner.
- A quarta regra instrui o serviço a excluir objetos na pasta Football que correspondem ao padrão *.ts depois que tiverem mais de 20 dias.
- A quinta regra instrui o serviço a excluir objetos na Football pasta que correspondam ao padrão index*.m3u8 depois de terem mais de 15 segundos. MediaStore exclui esses arquivos 16 segundos depois de serem colocados no contêiner.

- A sexta regra instrui o serviço a mover objetos na pasta Program para a classe de armazenamento IA após 30 dias de idade.

Para obter mais exemplos de políticas de ciclo de vida de objetos, consulte [Exemplos de política de ciclo de vida de objetos](#).

Adicionar uma política de ciclo de vida de objetos a um contêiner

Uma política de ciclo de vida de objetos permite especificar por quanto tempo armazenar seus objetos em um contêiner. Você define uma data de expiração e, após a data de expiração, o AWS Elemental MediaStore exclui os objetos. Leva até 20 minutos para o serviço aplicar a nova política ao contêiner.

Para obter informações sobre como criar uma política de ciclo de vida, consulte [Componentes de uma política de ciclo de vida de objetos](#).

Note

Para excluir regras de objetos (objetos expiram em dias), pode haver um pequeno atraso entre a expiração de um objeto e a exclusão do objeto. No entanto, as alterações no faturamento ocorrem assim que o objeto expira. Por exemplo, se uma regra de ciclo de vida especificar `10 days_since_create`, a conta não será cobrada pelo objeto depois que o objeto tiver 10 dias, mesmo que ele ainda não tenha sido excluído.

Como adicionar uma política de ciclo de vida de objetos (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), selecione o nome do contêiner para o qual você deseja criar uma política de ciclo de vida de objetos.

A página de detalhes do contêiner é exibida.

3. Na seção Object lifecycle policy (Política de ciclo de vida de objetos) escolha Create object lifecycle policy (Criar política de ciclo de vida de objetos).
4. Insira a política no formato JSON e, em seguida, selecione Save (Salvar).

Como adicionar uma política de ciclo de vida de objetos (AWS CLI)

1. Crie um arquivo que define a 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. No AWS CLI, use o `put-lifecycle-policy` comando:

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

Este comando não tem valor de retorno. O serviço anexa a política especificada ao contêiner.

Visualizar uma política de ciclo de vida de objetos

Uma política de ciclo de vida de objetos especifica por quanto tempo os objetos devem ser mantidos em um contêiner.

Como visualizar uma política de ciclo de vida de objetos (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), selecione o nome do contêiner para o qual você deseja visualizar a política de ciclo de vida de objetos.

A página de detalhes do contêiner é exibida, com a política de ciclo de vida de objetos na seção Object lifecycle policy (Política de ciclo de vida de objetos).

Para visualizar uma política de ciclo de vida de objetos (AWS CLI)

- No AWS CLI, use o `get-lifecycle-policy` comando:

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

O exemplo a seguir mostra o valor de retorno:

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

Editar uma política de ciclo de vida de objetos

Você não pode editar uma política de ciclo de vida de objetos existente. No entanto, você pode alterar uma política existente fazendo upload de uma política de substituição. Leva até 20 minutos para o serviço aplicar a política atualizada ao contêiner.

Como editar uma política de ciclo de vida de objetos (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), selecione o nome do contêiner para o qual você deseja editar a política de ciclo de vida de objetos.

A página de detalhes do contêiner é exibida.

3. Na seção Object lifecycle policy (Política de ciclo de vida de objetos), escolha Edit object lifecycle policy (Editar política de ciclo de vida de objetos).
4. Faça as alterações na política e selecione Save (Salvar).

Como editar uma política de ciclo de vida de objetos (AWS CLI)

1. Crie um arquivo que define a política de ciclo de vida do objeto atualizada:

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

2. No AWS CLI, use o `put-lifecycle-policy` comando:

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

Este comando não tem valor de retorno. O serviço anexa a política especificada ao contêiner, substituindo a política anterior.

Excluir uma política de ciclo de vida de objetos

Quando você exclui uma política de ciclo de vida de objeto, leva até 20 minutos para o serviço aplicar a alteração ao contêiner.

Como excluir uma política de ciclo de vida de objetos (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), selecione o nome do contêiner para o qual você deseja excluir a política de ciclo de vida de objetos.

A página de detalhes do contêiner é exibida.

3. Na seção Object lifecycle policy (Política de ciclo de vida de objetos), escolha Delete lifecycle policy (Excluir política de ciclo de vida de objetos).
4. Escolha Continue (Continuar) para confirmar e escolha Save (Salvar).

Para excluir uma política de ciclo de vida de objetos (AWS CLI)

- No AWS CLI, use o `delete-lifecycle-policy` comando:

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

Este comando não tem valor de retorno.

Exemplos de política de ciclo de vida de objetos

Os exemplos a seguir mostram políticas de ciclo de vida de objetos.

Tópicos

- [Exemplo de política de ciclo de vida de objetos: Expirar em segundos](#)
- [Exemplo de política de ciclo de vida de objetos: Expirar em dias](#)
- [Exemplo de política de ciclo de vida de objetos: transição para classe de armazenamento de acesso infrequente](#)
- [Exemplo de política de ciclo de vida de objetos: Várias regras](#)
- [Exemplo de política de ciclo de vida de objetos: Esvaziar contêiner](#)

Exemplo de política de ciclo de vida de objetos: Expirar em segundos

A política a seguir especifica que MediaStore exclui objetos que correspondam a todos os critérios a seguir:

- O objeto é adicionado ao contêiner após a política entrar em vigor.
- O objeto é armazenado na pasta Football.
- O objeto tem uma extensão de arquivo de m3u8.
- O objeto está no contêiner por mais de 20 segundos.

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

Exemplo de política de ciclo de vida de objetos: Expirar em dias

A política a seguir especifica que MediaStore exclui objetos que correspondam a todos os critérios a seguir:

- O objeto é armazenado na pasta Program
- O objeto tem uma extensão de arquivo de ts
- O objeto está no contêiner há mais de 5 dias

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

Exemplo de política de ciclo de vida de objetos: transição para classe de armazenamento de acesso infrequente

A política a seguir especifica que os objetos são MediaStore movidos para a classe de armazenamento de acesso infrequente (IA) quando eles tiverem 30 dias. Os objetos armazenados na classe de armazenamento IA têm taxas de armazenamento e recuperação diferentes dos objetos armazenados na classe de armazenamento padrão.

O campo `days_since_create` deve ser definido como `"numeric": [">=" , 30]`.

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

```

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

```

Exemplo de política de ciclo de vida de objetos: Várias regras

A política a seguir especifica que MediaStore faz o seguinte:

- Mova objetos armazenados na pasta AwardsShow para a classe de armazenamento de acesso infrequente (IA) após 30 dias
- Exclua objetos com a extensão de arquivo m3u8 e que estejam armazenados na pasta Football após 20 segundos
- Exclua objetos armazenados na pasta April após 10 dias
- Exclua objetos com a extensão de arquivo ts e que estejam armazenados na pasta Program após 5 dias

```

{
  "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"
  }
]
}

```

Exemplo de política de ciclo de vida de objetos: Esvaziar contêiner

A política de ciclo de vida do objeto a seguir especifica MediaStore que todos os objetos no contêiner, incluindo pastas e subpastas, 1 dia após serem adicionados ao contêiner. Se o contêiner contiver algum objeto antes da aplicação dessa política, MediaStore excluirá os objetos 1 dia após a política entrar em vigor. Leva até 20 minutos para o serviço aplicar a nova política ao contêiner.

```

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

```

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

Políticas métricas no AWS Elemental MediaStore

Para cada contêiner, você pode adicionar uma política métrica para permitir que o AWS Elemental envie métricas MediaStore para a Amazon CloudWatch. Pode demorar até 20 minutos para que a nova política tenha efeito. Para obter uma descrição de cada MediaStore métrica, consulte [MediaStore métricas](#).

Uma política de métrica contém o seguinte:

- Uma configuração para habilitar ou desabilitar métricas no nível do contêiner.
- De zero a cinco regras que habilitam métricas no nível do objeto. Se a política contiver regras, cada regra deverá incluir ambas as seguintes opções:
 - Um grupo de objetos que define quais objetos devem ser incluídos no grupo. A definição pode ser um caminho ou um nome de arquivo, mas não pode ter mais de 900 caracteres. Os caracteres válidos são: a–z, A–Z, 0–9, _ (sublinhado), = (igual), : (dois-pontos), . (ponto), - (hífen), ~ (til), / (barra) e * (asterisco). Curingas (*) são aceitáveis.
 - Um nome de grupo de objetos que permita fazer referência ao grupo de objetos. O nome não pode ter mais de 30 caracteres. Os caracteres válidos são a–z, A–Z, 0–9 e _ (sublinhado).

Se um objeto corresponder a várias regras, CloudWatch exibirá um ponto de dados para cada regra correspondente. Por exemplo, se um objeto corresponder a duas regras chamadas `rule1` e `rule2`, CloudWatch exibe dois pontos de dados para essas regras. O primeiro tem uma dimensão de `ObjectGroupName=rule1` e o segundo tem uma dimensão de `ObjectGroupName=rule2`.

Tópicos

- [Adicionar uma política de métrica](#)
- [Visualizar uma política de métrica](#)
- [Editar uma política de métrica](#)

- [Exemplos de políticas de métrica](#)

Adicionar uma política de métrica

Uma política métrica contém regras que determinam quais métricas o AWS MediaStore Elemental envia para a Amazon. CloudWatch Para obter exemplos de políticas de métrica, consulte [Exemplos de políticas de métrica](#).

Como adicionar uma política de métrica (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), escolha o nome do contêiner ao qual você deseja adicionar uma política de métrica.

A página de detalhes do contêiner é exibida.

3. Na seção Metric policy (Política de métrica), escolha Create metric policy (Criar política de métrica).
4. Insira a política no formato JSON e, em seguida, selecione Save (Salvar).

Visualizar uma política de métrica

Você pode usar o console ou o AWS CLI para visualizar a política métrica de um contêiner.

Como visualizar uma política de métrica (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), escolha o nome do contêiner.

A página de detalhes do contêiner é exibida. A política é exibida na seção Metric policy (Política de métrica).

Editar uma política de métrica

Uma política métrica contém regras que determinam quais métricas o AWS MediaStore Elemental envia para a Amazon. CloudWatch Quando você edita uma política de métrica existente, leva até 20 minutos para que a nova política entre em vigor. Para obter exemplos de políticas de métrica, consulte [Exemplos de políticas de métrica](#).

Como editar uma política de métrica (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), escolha o nome do contêiner.
3. Na seção Metric policy (Política de métrica), escolha Edit metric policy (Editar política de métrica).
4. Faça as alterações apropriadas e, em seguida, selecione Save (Salvar).

Exemplos de políticas de métrica

Os exemplos a seguir mostram políticas de métrica criadas para diferentes casos de uso.

Tópicos

- [Exemplo de política de métrica: métricas em nível de contêiner](#)
- [Exemplo de política métrica: métricas em nível de caminho](#)
- [Exemplo de política de métrica: métricas em nível de contêiner e de caminho](#)
- [Exemplo de política métrica: métricas em nível de caminho usando curingas](#)
- [Exemplo de política métrica: métricas em nível de caminho com regras sobrepostas](#)

Exemplo de política de métrica: métricas em nível de contêiner

Esse exemplo de política indica que o AWS Elemental MediaStore deve enviar métricas para a Amazon CloudWatch no nível do contêiner. Por exemplo, isso inclui a métrica RequestCount que conta o número de solicitações Put feitas ao contêiner. Como alternativa, é possível definir isso como DISABLED.

Como não há regras nessa política, MediaStore não envia métricas no nível do caminho. Por exemplo, não é possível ver quantas solicitações Put foram feitas para uma pasta específica dentro desse contêiner.

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

Exemplo de política métrica: métricas em nível de caminho

Esse exemplo de política indica que o AWS Elemental não MediaStore deve enviar métricas para a Amazon CloudWatch no nível do contêiner. Além disso, o MediaStore deve enviar métricas para objetos em duas pastas específicas: `baseball/saturday` e `football/saturday`. As métricas para solicitações do MediaStore são as seguintes:

- As solicitações para a `baseball/saturday` pasta têm uma CloudWatch dimensão `deObjectGroupName=baseballGroup`.
- As solicitações para a pasta `football/saturday` têm uma dimensão `ObjectGroupName=footballGroup`.

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

Exemplo de política de métrica: métricas em nível de contêiner e de caminho

Esse exemplo de política indica que o AWS Elemental MediaStore deve enviar métricas para a Amazon CloudWatch no nível do contêiner. Além disso, MediaStore deve enviar métricas para objetos em duas pastas específicas: `baseball/saturday` `football/saturday` e. As métricas para solicitações do MediaStore são as seguintes:

- As solicitações para a `baseball/saturday` pasta têm uma CloudWatch dimensão `deObjectGroupName=baseballGroup`.
- As solicitações para a `football/saturday` pasta têm uma CloudWatch dimensão `ObjectGroupName=footballGroup`.

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

Exemplo de política métrica: métricas em nível de caminho usando curingas

Esse exemplo de política indica que o AWS Elemental MediaStore deve enviar métricas para a Amazon CloudWatch no nível do contêiner. Além disso, também MediaStore deve enviar métricas para objetos com base no nome do arquivo. Um caractere curinga indica que os objetos podem ser armazenados em qualquer lugar do contêiner e podem ter qualquer nome de arquivo, contanto que ele termine com uma extensão `.m3u8`.

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

Exemplo de política métrica: métricas em nível de caminho com regras sobrepostas

Esse exemplo de política indica que o AWS Elemental MediaStore deve enviar métricas para a Amazon CloudWatch no nível do contêiner. Além disso, MediaStore deve enviar métricas para duas pastas: `sports/football/saturday` e `sports/football`.

As métricas das MediaStore solicitações para a `sports/football/saturday` pasta têm uma CloudWatch dimensão de `ObjectGroupName=footballGroup1`. Como os objetos armazenados na pasta `sports/football` correspondem a ambas as regras, o CloudWatch exibe dois pontos de

dados para esses objetos: um com uma dimensão de `ObjectGroupName=footballGroup1` e o segundo com uma dimensão de `ObjectGroupName=footballGroup2`.

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

Pastas no AWS Elemental MediaStore

As pastas são divisões dentro de um contêiner. Você usa as pastas para subdividir o contêiner da mesma maneira que você cria subpastas para dividir uma pasta em um sistema de arquivos. Você pode criar até 10 níveis de pastas, não incluindo o próprio contêiner.

As pastas são opcionais. Você pode optar por fazer upload de seus objetos diretamente para um contêiner em vez de uma pasta. No entanto, as pastas são uma maneira fácil de organizar os objetos.

Para fazer upload de um objeto para uma pasta, você especifica o caminho para a pasta. Se a pasta já existir, o AWS Elemental MediaStore armazena o objeto na pasta. Se a pasta não existir, o serviço a cria e, em seguida, armazena o objeto na pasta.

Por exemplo, suponha que você tenha um contêiner chamado `movies` e faça upload de um arquivo chamado `m1aw.ts` com o caminho `premium/canada`. O AWS Elemental MediaStore armazena o objeto na subpasta `canada` sob a pasta `premium`. Se nenhuma das pastas existir, o serviço criará a pasta `premium` e a subpasta `canada` e, em seguida, armazenará seu objeto na subpasta `canada`. Se você especificar apenas o contêiner `movies` (sem caminho), o serviço armazenará o objeto diretamente no contêiner.

O AWS Elemental exclui MediaStore automaticamente uma pasta quando você exclui o último objeto dessa pasta. O serviço também exclui todas as pastas vazias acima dessa pasta. Por exemplo, suponha que você tenha uma pasta chamada `premium` que não contenha nenhum arquivo, mas contenha uma subpasta denominada `canada`. A subpasta `canada` contém um arquivo chamado `m1aw.ts`. Se você excluir o arquivo `m1aw.ts`, o serviço excluirá as pastas `premium` e `canada`. Essa exclusão automática se aplica apenas a pastas. O serviço não exclui contêineres vazios.

Tópicos

- [Regras para nomes de pastas](#)
- [Criar uma pasta](#)
- [Excluir uma pasta](#)

Regras para nomes de pastas

Ao escolher um nome para a pasta, lembre-se do seguinte:

- O nome pode conter apenas os seguintes caracteres: letras maiúsculas (A-Z), letras minúsculas (a-z), números (0-9), pontos finais (.), hífens (-), tildes (~), sublinhados (_), sinais de igualdade (=) e dois-pontos (:).
- O nome deve ter pelo menos um caractere. Nomes de pastas vazias (como `folder1//folder3/`) não são permitidos.
- Os nomes diferenciam letras maiúsculas de minúsculas. Por exemplo, você pode ter uma pasta chamada `myFolder` e outra chamada `myfolder` no mesmo contêiner ou pasta porque esses nomes são exclusivos.
- O nome deve ser exclusivo apenas dentro do contêiner ou da pasta pai. Por exemplo, você pode criar uma pasta chamada `myfolder` em dois contêineres diferentes: `movies/myfolder` e `sports/myfolder`.
- O nome pode ter o mesmo nome que o contêiner pai.
- A pasta não pode ser renomeada depois de criada.

Criar uma pasta

Você pode criar pastas ao fazer upload de objetos. Para fazer upload de um objeto para uma pasta, você especifica o caminho para a pasta. Se a pasta já existir, o AWS Elemental MediaStore armazena o objeto na pasta. Se a pasta não existir, o serviço a cria e, em seguida, armazena o objeto na pasta.

Para obter mais informações, consulte [the section called “Fazer upload de um objeto”](#).

Excluir uma pasta

Você pode excluir pastas somente se estiverem vazias, não é possível excluir pastas que contêm objetos.

O AWS Elemental exclui MediaStore automaticamente uma pasta quando você exclui o último objeto dessa pasta. O serviço também exclui todas as pastas vazias acima dessa pasta. Por exemplo, suponha que você tenha uma pasta chamada `premium` que não contenha nenhum arquivo, mas contenha uma subpasta denominada `canada`. A subpasta `canada` contém um arquivo chamado `m1aw.ts`. Se você excluir o arquivo `m1aw.ts`, o serviço excluirá as pastas `premium` e `canada`. Essa exclusão automática se aplica apenas a pastas. O serviço não exclui contêineres vazios.

Para obter mais informações, consulte [Excluir um objeto](#).

Objetos no AWS Elemental MediaStore

Os MediaStore ativos do AWS Elemental são chamados de objetos. Você pode fazer upload de um objeto para um contêiner ou para uma pasta no contêiner.

Em MediaStore, você pode carregar, baixar e excluir objetos:

- Upload (Fazer upload): adicionar um objeto a um contêiner ou uma pasta. Isso não é o mesmo que criar um objeto. Você deve criar seus objetos localmente antes de poder carregá-los no MediaStore.
- Baixar — Copie um objeto MediaStore de outro local. Isso não remove o objeto de MediaStore.
- Excluir — remover completamente um objeto do MediaStore. Você pode excluir objetos individualmente, ou você pode [adicionar uma política de ciclo de vida de objetos](#) para excluir automaticamente objetos dentro de um contêiner após uma duração especificada.

MediaStore aceita todos os tipos de arquivo.

Tópicos

- [Fazer upload de um objeto](#)
- [Visualizar uma lista de objetos](#)
- [Visualizar os detalhes de um objeto](#)
- [Fazer download de um objeto](#)
- [Excluir objetos](#)

Fazer upload de um objeto

Você pode fazer upload de objetos para um contêiner ou para uma pasta em um contêiner. Para fazer upload de um objeto para uma pasta, você especifica o caminho para a pasta. Se a pasta já existir, o AWS Elemental MediaStore armazena o objeto na pasta. Se a pasta não existir, o serviço a cria e, em seguida, armazena o objeto na pasta. Para obter mais informações sobre pastas, consulte [Pastas no AWS Elemental MediaStore](#).

Você pode usar o MediaStore console ou o AWS CLI para carregar objetos.

MediaStore oferece suporte à transferência fragmentada de objetos, o que reduz a latência ao disponibilizar um objeto para download enquanto ele ainda está sendo carregado. Para usar esse

recurso, defina a disponibilidade de upload do objeto como `streaming`. Você pode definir o valor desse cabeçalho ao [fazer upload do objeto usando a API](#). Se você não especificar esse cabeçalho em sua solicitação, MediaStore atribuirá o valor padrão de `standard` para a disponibilidade de upload do objeto.

Os tamanhos de objeto não podem exceder 25 MB para disponibilidade padrão de upload e 10 MB para disponibilidade de upload de streaming.

Note

Os nomes de arquivo de objeto podem conter apenas letras, números, pontos (.), sublinhados (_), tildes (~), hifens (-), sinais de igual (=) e dois pontos (:).

Para fazer upload de um objeto (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), selecione o nome do contêiner. O painel de detalhes do contêiner é exibido.
3. Selecione Upload object (Fazer upload de objeto).
4. Em Target path (Caminho de destino), digite o caminho das pastas. Por exemplo, `.premium/canada`. Se alguma das pastas no caminho especificado ainda não existir, o serviço as criará automaticamente.
5. Na seção Object (Objeto), selecione Browse (Navegar).
6. Navegue até a pasta apropriada e escolha um objeto para fazer upload.
7. Selecione Open (Abrir) e Upload (Fazer upload).

Note

Se um arquivo com o mesmo nome já existir na pasta selecionada, o serviço substituirá o arquivo original pelo arquivo carregado.

Para carregar um objeto (AWS CLI)

- No AWS CLI, use o `put-object` comando. Também é possível incluir qualquer um dos seguintes parâmetros: `content-type`, `cache-control` (para permitir que o chamador

controle o comportamento de cache do objeto) e path (para colocar o objeto em uma pasta dentro do contêiner).

Note

Após fazer upload do objeto, você poderá editar o `content-type`, `cache-control` ou `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
```

O exemplo a seguir mostra o valor de retorno:

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

Visualizar uma lista de objetos

Você pode usar o MediaStore console do AWS Elemental para visualizar itens (objetos e pastas) armazenados no nível mais alto de um contêiner ou em uma pasta. Os itens armazenados em uma subpasta do contêiner ou da pasta atual não serão exibidos. Você pode usar o AWS CLI para visualizar uma lista de objetos e pastas em um contêiner, independentemente de quantas pastas ou subpastas estejam dentro do contêiner.

Para visualizar uma lista de objetos em um contêiner específico (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), selecione o nome do contêiner que tem a pasta que você deseja visualizar.
3. Escolha o nome da pasta na lista.

Uma página de detalhes é exibida, mostrando todas as pastas e objetos que estão armazenados na pasta.

Para visualizar uma lista de objetos em uma pasta específica (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), selecione o nome do contêiner que tem a pasta que você deseja visualizar.

Uma página de detalhes é exibida, mostrando todas as pastas e objetos que estão armazenados no contêiner.

Para visualizar uma lista de objetos e pastas em um contêiner específico (AWS CLI)

- No AWS CLI, use o `list-items` comando:

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

O exemplo a seguir mostra o valor de retorno:

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

Note

Os objetos que estão sujeitos a uma regra `seconds_since_create` não são incluídos em uma resposta `list-items`.

Para visualizar uma lista de objetos e pastas em uma pasta específica (AWS CLI)

- No AWS CLI, use o `list-items` comando, com o nome da pasta especificada no final da solicitação:

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

O exemplo a seguir mostra o 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

Os objetos que estão sujeitos a uma regra `seconds_since_create` não são incluídos em uma resposta `list-items`.

Visualizar os detalhes de um objeto

Depois de fazer o upload de um objeto, o AWS Elemental MediaStore armazena detalhes como data de modificação, tamanho do conteúdo ETag (tag de entidade) e tipo de conteúdo. Para saber como os metadados de um objeto são usados, consulte [Interação do MediaStore com caches HTTP](#).

Para visualizar os detalhes de um objeto (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), selecione o nome do contêiner que tem o objeto que você deseja visualizar.
3. Se o objeto do qual você deseja visualizar estiver em uma pasta, continue escolhendo os nomes das pastas até ver o objeto.
4. Escolha o nome do objeto.

Uma página de detalhes será exibida, mostrando informações sobre o objeto.

Para visualizar os detalhes de um objeto (AWS CLI)

- No AWS CLI, use o `describe-object` comando:

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

O exemplo a seguir mostra o valor de retorno:

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


Note

Quando você exclui o único objeto em uma pasta, o AWS Elemental exclui MediaStore automaticamente a pasta e todas as pastas vazias acima dessa pasta. Por exemplo, suponha que você tenha uma pasta chamada `premium` que não contenha nenhum arquivo, mas contenha uma subpasta denominada `canada`. A subpasta `canada` contém um arquivo chamado `m1aw.ts`. Se você excluir o arquivo `m1aw.ts`, o serviço excluirá as pastas `premium` e `canada`.

Para excluir um objeto (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), selecione o nome do contêiner que tem o objeto que você deseja excluir.
3. Se o objeto que você deseja excluir estiver em uma pasta, continue escolhendo os nomes das pastas até ver o objeto.
4. Escolha a opção à esquerda do nome do objeto.
5. Escolha Excluir.

Para excluir um objeto (AWS CLI)

- No AWS CLI, use o `delete-object` comando.

Exemplo:

```
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 não tem valor de retorno.

Esvaziar um contêiner

É possível esvaziar um contêiner para excluir todos os objetos armazenados nele. Como alternativa, é possível adicionar uma [política de ciclo de vida do objeto](#) para excluir objetos automaticamente depois que eles atingem uma certa idade em um contêiner, ou [excluir objetos individualmente](#).

Como esvaziar um contêiner (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), escolha a opção para o contêiner que você deseja esvaziar.
3. Escolha Empty container (Esvaziar contêiner). Uma mensagem de confirmação será exibida.
4. Confirme se deseja esvaziar o contêiner inserindo o nome do contêiner no campo de texto e escolha Esvaziar.

Segurança no AWS Elemental MediaStore

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de data centers e arquiteturas de rede criados para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como segurança da nuvem e segurança na nuvem:

- **Segurança da nuvem** — AWS é responsável por proteger a infraestrutura que executa AWS os serviços no Nuvem AWS. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de [AWS](#) de . Para saber mais sobre os programas de conformidade que se aplicam ao AWS Elemental MediaStore, consulte [AWS Serviços no escopo por programa de conformidade AWS Serviços em Escopo por programa](#) .
- **Segurança na nuvem** — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar MediaStore. Os tópicos a seguir mostram como configurar para atender MediaStore aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus MediaStore recursos.

Tópicos

- [Proteção de dados no AWS Elemental MediaStore](#)
- [Identity and Access Management para AWS Elemental MediaStore](#)
- [Registro e monitoramento em AWS Elemental MediaStore](#)
- [Validação de conformidade para o AWS Elemental MediaStore](#)
- [Resiliência no AWS Elemental MediaStore](#)
- [Segurança da infraestrutura no AWS Elemental MediaStore](#)
- [Prevenção contra o ataque do “substituto confuso” em todos os serviços](#)

Proteção de dados no AWS Elemental MediaStore

O [modelo de responsabilidade AWS compartilhada](#) se aplica à proteção de dados no AWS Elemental MediaStore. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre a privacidade de dados, consulte as [Data Privacy FAQ](#). Para obter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog [AWS Shared Responsibility Model and RGPD](#) no Blog de segurança da AWS .

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como [trabalhar com CloudTrail trilhas](#) no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para obter mais informações sobre os endpoints FIPS disponíveis, consulte [Federal Information Processing Standard \(FIPS\) 140-3](#).

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sigilosas, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com MediaStore ou Serviços da AWS usa o console, a API ou AWS SDKs. AWS CLI Quaisquer dados inseridos em tags ou em campos de texto de formato livre usados

para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, recomendamos fortemente que não inclua informações relacionadas às credenciais no URL para validar sua solicitação para esse servidor.

Criptografia de dados

MediaStore criptografa contêineres e objetos em repouso usando o algoritmo AES-256 padrão do setor. Recomendamos que você use MediaStore para proteger seus dados das seguintes maneiras:

- Crie uma política de contêiner para controlar os direitos de acesso a todas as pastas e objetos desse contêiner. Para obter mais informações, consulte [the section called “Políticas de contêiner”](#).
- Crie uma política de compartilhamento de recursos de origem cruzada (CORS) para permitir o acesso de origem cruzada seletivamente aos seus recursos. MediaStore Com o CORS, você pode permitir que aplicativos web de clientes carregados em um domínio interajam com recursos em outro domínio. Para obter mais informações, consulte [the section called “Políticas de CORS”](#).

Identity and Access Management para AWS Elemental MediaStore

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar MediaStore os recursos. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

- [Público](#)
- [Autenticação com identidades](#)
- [Gerenciar o acesso usando políticas](#)
- [Como o AWS Elemental MediaStore funciona com o IAM](#)
- [Exemplos de políticas baseadas em identidade para o AWS Elemental MediaStore](#)
- [Solução de problemas de MediaStore identidade e acesso ao AWS Elemental](#)

Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz MediaStore.

Usuário do serviço — Se você usar o MediaStore serviço para realizar seu trabalho, seu administrador fornecerá as credenciais e as permissões de que você precisa. À medida que você usa mais MediaStore recursos para fazer seu trabalho, talvez precise de permissões adicionais. Entender como o acesso é gerenciado pode ajudá-lo a solicitar as permissões corretas ao seu administrador. Se não for possível acessar um recurso no MediaStore, consulte [Solução de problemas de MediaStore identidade e acesso ao AWS Elemental](#).

Administrador de serviços — Se você é responsável pelos MediaStore recursos da sua empresa, provavelmente tem acesso total MediaStore a. É seu trabalho determinar quais MediaStore recursos e recursos seus usuários do serviço devem acessar. Envie as solicitações ao administrador do IAM para alterar as permissões dos usuários de serviço. Revise as informações nesta página para compreender os conceitos básicos do IAM. Para saber mais sobre como sua empresa pode usar o IAM com MediaStore, consulte [Como o AWS Elemental MediaStore funciona com o IAM](#).

Administrador do IAM: se você for um administrador do IAM, talvez queira saber detalhes sobre como pode gravar políticas para gerenciar o acesso ao MediaStore. Para ver exemplos de políticas MediaStore baseadas em identidade que você pode usar no IAM, consulte. [Exemplos de políticas baseadas em identidade para o AWS Elemental MediaStore](#)

Autenticação com identidades

A autenticação é como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como o Usuário raiz da conta da AWS, como usuário do IAM ou assumindo uma função do IAM.

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Usuários (IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como identidade federada, o administrador já configurou anteriormente a federação de identidades usando perfis do IAM. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login em AWS, consulte [Como fazer login Conta da AWS](#) no Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para designar solicitações por conta própria, consulte [Versão 4 do AWS Signature para solicitações de API](#) no Guia do usuário do IAM.

Independente do método de autenticação usado, também pode ser necessário fornecer informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte [Autenticação multifator](#) no Guia do usuário do AWS IAM Identity Center e [Usar a autenticação multifator da AWS no IAM](#) no Guia do usuário do IAM.

Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário-raiz para tarefas diárias. Proteja as credenciais do usuário-raiz e use-as para executar as tarefas que somente ele puder executar. Para obter a lista completa das tarefas que exigem login como usuário-raiz, consulte [Tarefas que exigem credenciais de usuário-raiz](#) no Guia do Usuário do IAM.

Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório de usuários corporativo, de um provedor de identidade da web AWS Directory Service, do diretório do Identity Center ou de qualquer usuário que acesse usando credenciais fornecidas Serviços da AWS por meio de uma fonte de identidade. Quando as identidades federadas são acessadas Contas da AWS, elas assumem funções, e as funções fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, é recomendável usar o AWS IAM Identity Center. Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todos os seus Contas

da AWS aplicativos. Para obter mais informações sobre o Centro de Identidade do IAM, consulte [O que é o Centro de Identidade do IAM?](#) no Guia do Usuário do AWS IAM Identity Center .

Usuários e grupos do IAM

Um [usuário do IAM](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, é recomendável contar com credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, é recomendável alternar as chaves de acesso. Para obter mais informações, consulte [Alternar as chaves de acesso regularmente para casos de uso que exijam credenciais de longo prazo](#) no Guia do Usuário do IAM.

Um [grupo do IAM](#) é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdminse conceder a esse grupo permissões para administrar recursos do IAM.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte [Casos de uso para usuários do IAM](#) no Guia do usuário do IAM.

Perfis do IAM

Uma [função do IAM](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. Para assumir temporariamente uma função do IAM no AWS Management Console, você pode [alternar de um usuário para uma função do IAM \(console\)](#). Você pode assumir uma função chamando uma operação de AWS API AWS CLI ou usando uma URL personalizada. Para obter mais informações sobre métodos para usar perfis, consulte [Métodos para assumir um perfil](#) no Guia do usuário do IAM.

Perfis do IAM com credenciais temporárias são úteis nas seguintes situações:

- **Acesso de usuário federado:** para atribuir permissões a identidades federadas, é possível criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas por ele. Para ter mais informações sobre perfis para federação, consulte [Criar um perfil para um provedor de identidade de terceiros \(federação\)](#) no Guia do usuário do IAM. Se usar o Centro de Identidade do IAM,

configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o Centro de Identidade do IAM correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte [Conjuntos de Permissões](#) no Guia do Usuário do AWS IAM Identity Center .

- Permissões temporárias para usuários do IAM: um usuário ou um perfil do IAM pode presumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas: é possível usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para conhecer a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.
- Acesso entre serviços — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos na Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões da entidade principal da chamada, usando um perfil de serviço ou um perfil vinculado ao serviço.
- Sessões de acesso direto (FAS) — Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte [Sessões de acesso direto](#).
- Perfil de serviço: um perfil de serviço é um [perfil do IAM](#) que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte [Criar um perfil para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do Usuário do IAM.
- Função vinculada ao serviço — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um AWS service (Serviço da AWS) O serviço pode presumir o perfil para executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados a serviço.

- Aplicativos em execução na Amazon EC2 — Você pode usar uma função do IAM para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma EC2 instância e fazendo solicitações AWS CLI de AWS API. Isso é preferível a armazenar chaves de acesso na EC2 instância. Para atribuir uma AWS função a uma EC2 instância e disponibilizá-la para todos os aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém a função e permite que os programas em execução na EC2 instância recebam credenciais temporárias. Para obter mais informações, consulte [Usar uma função do IAM para conceder permissões a aplicativos executados em EC2 instâncias da Amazon](#) no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte [Visão geral das políticas JSON](#) no Guia do usuário do IAM.

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e perfis não têm permissões. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis e os usuários podem assumir os perfis.

As políticas do IAM definem permissões para uma ação independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função da AWS Management Console AWS CLI, da ou da AWS API.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que

condições. Para saber como criar uma política baseada em identidade, consulte [Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente](#) no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte [Escolher entre políticas gerenciadas e políticas em linha](#) no Guia do usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

O Amazon S3 e o AWS WAF Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber mais ACLs, consulte a [visão geral da lista de controle de acesso \(ACL\)](#) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- **Limites de permissões:** um limite de permissões é um recurso avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (usuário ou perfil do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte [Limites de permissões para identidades do IAM](#) no Guia do usuário do IAM.
- **Políticas de controle de serviço (SCPs)** — SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em AWS Organizations. AWS Organizations é um serviço para agrupar e gerenciar centralmente várias Contas da AWS que sua empresa possui. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as suas contas. O SCP limita as permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre Organizations e SCPs, consulte [Políticas de controle de serviços](#) no Guia AWS Organizations do Usuário.
- **Políticas de controle de recursos (RCPs)** — RCPs são políticas JSON que você pode usar para definir o máximo de permissões disponíveis para recursos em suas contas sem atualizar as políticas do IAM anexadas a cada recurso que você possui. O RCP limita as permissões para recursos nas contas dos membros e pode afetar as permissões efetivas para identidades, incluindo a Usuário raiz da conta da AWS, independentemente de pertencerem à sua organização. Para obter mais informações sobre Organizations e RCPs, incluindo uma lista Serviços da AWS desse suporte RCPs, consulte [Políticas de controle de recursos \(RCPs\)](#) no Guia AWS Organizations do usuário.
- **Políticas de sessão:** são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em recursos. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte [Políticas de sessão](#) no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte [Lógica de avaliação de políticas](#) no Guia do usuário do IAM.

Como o AWS Elemental MediaStore funciona com o IAM

Antes de usar o IAM para gerenciar o acesso MediaStore, saiba com quais recursos do IAM estão disponíveis para uso MediaStore.

Recursos do IAM que você pode usar com o AWS Elemental MediaStore

Recurso do IAM	MediaStore apoio
Políticas baseadas em identidade	Sim
Políticas baseadas em atributos	Sim
Ações de políticas	Sim
Recursos de políticas	Sim
Chaves de condição de política (específicas do serviço)	Sim
ACLs	Não
ABAC (tags em políticas)	Parcial
Credenciais temporárias	Sim
Permissões de entidade principal	Sim
Perfis de serviço	Sim
Perfis vinculados a serviço	Não

Para ter uma visão de alto nível de como MediaStore e outros AWS serviços funcionam com a maioria dos recursos do IAM, consulte [AWS os serviços que funcionam com o IAM](#) no Guia do usuário do IAM.

Políticas baseadas em identidade para MediaStore

Compatível com políticas baseadas em identidade: sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente](#) no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte [Referência de elemento de política JSON do IAM](#) no Guia do usuário do IAM.

Exemplos de políticas baseadas em identidade para MediaStore

Para ver exemplos de políticas MediaStore baseadas em identidade, consulte. [Exemplos de políticas baseadas em identidade para o AWS Elemental MediaStore](#)

Políticas baseadas em recursos dentro MediaStore

Compatível com políticas baseadas em recursos: sim

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para permitir o acesso entre contas, você pode especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recursos. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um administrador do IAM na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Consulte mais informações em [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.

Note

MediaStore também oferece suporte a políticas de contêiner que definem quais entidades principais (contas, usuários, funções e usuários federados) podem realizar ações no contêiner. Para obter mais informações, consulte [Políticas de contêiner](#).

Ações políticas para MediaStore

Compatível com ações de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da operação de AWS API associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de MediaStore ações, consulte [Ações definidas pelo AWS Elemental MediaStore na Referência](#) de autorização de serviço.

As ações de política MediaStore usam o seguinte prefixo antes da ação:

```
mediastore
```

Para especificar várias ações em uma única declaração, separe-as com vírgulas.

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

Para ver exemplos de políticas MediaStore baseadas em identidade, consulte [Exemplos de políticas baseadas em identidade para o AWS Elemental MediaStore](#)

Recursos políticos para MediaStore

Compatível com recursos de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou `NotResource`. Como prática recomendada, especifique um recurso usando seu [nome do recurso da Amazon \(ARN\)](#). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

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

Para ver uma lista dos tipos de MediaStore recursos e seus ARNs, consulte [Recursos definidos pelo AWS Elemental MediaStore](#) na Referência de Autorização de Serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte [Ações definidas pelo AWS Elemental MediaStore](#).

O recurso de MediaStore contêiner tem o seguinte ARN:

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

Para obter mais informações sobre o formato de ARNs, consulte [Amazon Resource Names \(ARNs\) e AWS Service Namespaces](#).

Por exemplo, para especificar o contêiner AwardsShow em sua instrução, use o seguinte ARN:

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

Chaves de condição de política para MediaStore

Compatível com chaves de condição de política específicas de serviço: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usem [agentes de condição](#), como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos de `Condition` em uma declaração ou várias chaves em um único elemento de `Condition`, a AWS os avaliará usando uma operação lógica AND. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma OR operação lógica. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um recurso somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte [Elementos da política do IAM: variáveis e tags](#) no Guia do usuário do IAM.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição AWS globais, consulte as [chaves de contexto de condição AWS global](#) no Guia do usuário do IAM.

Para ver uma lista de chaves de MediaStore condição, consulte Chaves de [condição para o AWS Elemental MediaStore na Referência](#) de autorização de serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte [Ações definidas pelo AWS Elemental MediaStore](#).

Para ver exemplos de políticas MediaStore baseadas em identidade, consulte [Exemplos de políticas baseadas em identidade para o AWS Elemental MediaStore](#)

ACLs in MediaStore

Suportes ACLs: Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

ABAC com MediaStore

Compatível com ABAC (tags em políticas): parcial

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define as permissões com base em atributos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a entidades do IAM (usuários ou funções) e a vários AWS recursos. Marcar de entidades e atributos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do recurso que ela estiver tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna um problema.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no [elemento de condição](#) de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

Se um serviço for compatível com as três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com as três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para obter mais informações sobre o ABAC, consulte [Definir permissões com autorização do ABAC](#) no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte [Usar controle de acesso baseado em atributos \(ABAC\)](#) no Guia do usuário do IAM.

Usando credenciais temporárias com MediaStore

Compatível com credenciais temporárias: sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte Serviços da AWS "[Trabalhe com o IAM](#)" no Guia do usuário do IAM.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre como alternar funções, consulte [Alternar para um perfil do IAM \(console\)](#) no Guia do usuário do IAM.

Você pode criar manualmente credenciais temporárias usando a AWS API AWS CLI ou. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para obter mais informações, consulte [Credenciais de segurança temporárias no IAM](#).

Permissões principais entre serviços para MediaStore

Compatibilidade com o recurso de encaminhamento de sessões de acesso (FAS): sim

Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte [Sessões de acesso direto](#).

Funções de serviço para MediaStore

Compatível com perfis de serviço: sim

O perfil de serviço é um [perfil do IAM](#) que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte [Criar um perfil para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do Usuário do IAM.

Warning

Alterar as permissões de uma função de serviço pode interromper MediaStore a funcionalidade. Edite as funções de serviço somente quando MediaStore fornecer orientação para fazer isso.

Funções vinculadas a serviços para MediaStore

Compatível com perfis vinculados ao serviço: Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um AWS service (Serviço da AWS). O serviço pode presumir o perfil para executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para funções vinculadas ao serviço.

Para obter detalhes sobre como criar ou gerenciar perfis vinculados a serviços, consulte [Serviços da AWS que funcionam com o IAM](#). Encontre um serviço na tabela que inclua um Yes na coluna Perfil vinculado ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.

Exemplos de políticas baseadas em identidade para o AWS Elemental MediaStore

Por padrão, usuários e perfis não têm permissão para criar ou modificar recursos do MediaStore. Eles também não podem realizar tarefas usando a AWS API, AWS Management Console, AWS Command Line Interface (AWS CLI) ou. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis e os usuários podem assumir os perfis.

Para aprender a criar uma política baseada em identidade do IAM ao usar esses documentos de política em JSON de exemplo, consulte [Criar políticas do IAM \(console\)](#) no Guia do usuário do IAM.

Para obter detalhes sobre ações e tipos de recursos definidos por MediaStore, incluindo o formato de cada um dos tipos de recursos, consulte [Ações, recursos e chaves de condição para o AWS Elemental MediaStore](#) na Referência de Autorização de Serviço. ARNs

Tópicos

- [Práticas recomendadas de política](#)
- [Usar o console do MediaStore](#)
- [Permitir que os usuários visualizem suas próprias permissões](#)

Práticas recomendadas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir MediaStore recursos em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para obter mais informações, consulte [Políticas gerenciadas pela AWS](#) ou [Políticas gerenciadas pela AWS para funções de trabalho](#) no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte [Políticas e permissões no IAM](#) no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como AWS CloudFormation. Para obter mais informações, consulte [Elementos da política JSON do IAM: condição](#) no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para obter mais informações, consulte [Validação de políticas do IAM Access Analyzer](#) no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte [Configuração de acesso à API protegido por MFA](#) no Guia do Usuário do IAM.

Para obter mais informações sobre as práticas recomendadas do IAM, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Usar o console do MediaStore

Para acessar o MediaStore console AWS Elemental, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os MediaStore recursos em seu Conta da AWS. Caso crie uma política baseada em identidade mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente a ações que correspondam à operação de API que estiverem tentando executar.

Para garantir que usuários e funções ainda possam usar o MediaStore console, anexe também a política MediaStore *ConsoleAccess* ou a política *ReadOnly* AWS gerenciada às entidades. Para obter informações, consulte [Adicionar permissões a um usuário](#) no Guia do usuário do IAM.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou AWS .

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

Solução de problemas de MediaStore identidade e acesso ao AWS Elemental

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com MediaStore um IAM.

Tópicos

- [Não estou autorizado a realizar uma ação em MediaStore](#)
- [Não estou autorizado a realizar iam: PassRole](#)
- [Quero permitir que pessoas fora da minha Conta da AWS acessem meus MediaStore recursos](#)

Não estou autorizado a realizar uma ação em MediaStore

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, suas políticas deverão ser atualizadas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário do IAM `mateojackson` tenta usar o console para visualizar detalhes sobre um atributo `my-example-widget` fictício, mas não tem as permissões `mediastore:GetWidget` fictícias.

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

Nesse caso, a política do usuário mateojackson deve ser atualizada para permitir o acesso ao recurso *my-example-widget* usando a ação *mediastore:GetWidget*.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Não estou autorizado a realizar iam: PassRole

Se você receber uma mensagem de erro informando que não está autorizado a executar a ação *iam:PassRole*, as suas políticas devem ser atualizadas para permitir que você passe uma função para o MediaStore.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazê-lo, você deve ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando uma usuária do IAM chamada *marymajor* tenta utilizar o console para executar uma ação no MediaStore. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

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

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação *iam:PassRole*.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas fora da minha Conta da AWS acessem meus MediaStore recursos

Você pode criar um perfil que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para

serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se é MediaStore compatível com esses recursos, consulte [Como o AWS Elemental MediaStore funciona com o IAM](#).
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como [fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui](#) no Guia do usuário do IAM.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como [fornecer acesso Contas da AWS a terceiros](#) no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte [Conceder acesso a usuários autenticados externamente \(federação de identidades\)](#) no Guia do usuário do IAM.
- Para saber a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.

Registro e monitoramento em AWS Elemental MediaStore

Esta seção fornece uma visão geral das opções de registro em log e monitoramento no AWS Elemental MediaStore para fins de segurança. Para obter mais informações sobre o registro e o monitoramento no MediaStore consulte [Monitoramento e marcação no AWS Elemental MediaStore](#).

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho de AWS Elemental MediaStore suas AWS soluções. Você deve coletar dados de monitoramento de todas as partes da sua AWS solução para poder depurar com mais facilidade uma falha multiponto, caso ocorra. AWS fornece várias ferramentas para monitorar seus MediaStore recursos e responder a possíveis incidentes.

CloudWatch Alarmes da Amazon

Usando CloudWatch alarmes, você observa uma única métrica durante um período de tempo especificado por você. Se a métrica exceder um determinado limite, uma notificação será enviada para um tópico do Amazon SNS ou para uma política do AWS Auto Scaling. CloudWatch os alarmes não invocam ações porque estão em um estado específico. O estado deve ter sido alterado e

mantido por uma quantidade especificada de períodos. Para obter mais informações, consulte [Monitoramento com CloudWatch](#).

AWS CloudTrail troncos

CloudTrail fornece um registro das ações realizadas por um usuário, função ou AWS serviço em AWS Elemental MediaStore. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita MediaStore, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais. Para obter mais informações, consulte [Registrar chamadas de API com o CloudTrail](#).

AWS Trusted Advisor

Trusted Advisor baseia-se nas melhores práticas aprendidas ao atender centenas de milhares de AWS clientes. Trusted Advisor inspeciona seu ambiente da AWS e, em seguida, faz recomendações quando existem oportunidades para economizar dinheiro, melhorar a disponibilidade e o desempenho do sistema ou ajudar a fechar lacunas de segurança. Todos os AWS clientes têm acesso a cinco cheques do Trusted Advisor. Clientes com um plano de suporte Business ou Enterprise podem ver todos os Trusted Advisor cheques.

Para obter mais informações, consulte [AWS Trusted Advisor](#).

Validação de conformidade para o AWS Elemental MediaStore

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte [Serviços da AWS Escopo por Programa de Conformidade](#) [Serviços da AWS](#) e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de [AWS conformidade Programas AWS](#) de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte [Baixar relatórios em AWS Artifact](#) .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- [Governança e conformidade de segurança](#): esses guias de implementação de solução abordam considerações sobre a arquitetura e fornecem etapas para implantar recursos de segurança e conformidade.

- [Referência de serviços qualificados para HIPAA](#): lista os serviços qualificados para HIPAA. Nem todos Serviços da AWS são elegíveis para a HIPAA.
- AWS Recursos de <https://aws.amazon.com/compliance/resources/> de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- [AWS Guias de conformidade do cliente](#) — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização (ISO)).
- [Avaliação de recursos com regras](#) no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- [AWS Security Hub](#)— Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os recursos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a [Referência de controles do Security Hub](#).
- [Amazon GuardDuty](#) — Isso AWS service (Serviço da AWS) detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, como o PCI DSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.
- [AWS Audit Manager](#)— Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

Resiliência no AWS Elemental MediaStore

A infraestrutura AWS global é construída em torno Regiões da AWS de zonas de disponibilidade. Regiões da AWS fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre zonas de disponibilidade Regiões da AWS e zonas de disponibilidade, consulte [Infraestrutura AWS global](#).

Além da infraestrutura AWS global, MediaStore oferece vários recursos para ajudar a suportar suas necessidades de resiliência e backup de dados.

Segurança da infraestrutura no AWS Elemental MediaStore

Como um serviço gerenciado, o AWS Elemental MediaStore é protegido pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte [AWS Cloud Security](#). Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte [Proteção](#) de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa chamadas de API AWS publicadas para acessar MediaStore pela rede. Os clientes devem oferecer compatibilidade com:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou é possível usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Prevenção contra o ataque do “substituto confuso” em todos os serviços

“Confused deputy” é um problema de segurança no qual uma entidade sem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Em AWS, a falsificação de identidade entre serviços pode resultar no problema confuso do deputado. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar.

Para evitar isso, a AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta.

Recomendamos usar as chaves de contexto de condição [aws:SourceAccount](#) global [aws:SourceArn](#) as chaves de contexto nas políticas de recursos para limitar as permissões que o AWS Elemental MediaStore concede a outro serviço ao recurso. Use `aws:SourceArn` se quiser que apenas um recurso seja associado ao acesso entre serviços. Use `aws:SourceAccount` se quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços.

A maneira mais eficaz de se proteger contra o problema do substituto confuso é usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso. Se você não souber o ARN completo do recurso ou especificar vários recursos, use a chave de condição de contexto global `aws:SourceArn` com caracteres curinga (*) para as partes desconhecidas do ARN. Por exemplo, `.arn:aws:service:*:123456789012:*`

Se o valor de `aws:SourceArn` não contiver o ID da conta, como um ARN de bucket do Amazon S3, você deverá usar ambas as chaves de contexto de condição global para limitar as permissões.

O valor de `aws:SourceArn` deve ser a configuração que MediaStore publica CloudWatch registros em sua região e conta.

O exemplo a seguir mostra como você pode usar as chaves de contexto de condição `aws:SourceAccount` global `aws:SourceArn` e as chaves de contexto MediaStore para evitar o confuso problema substituto.

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

```
"StringEquals": {  
  "aws:SourceAccount": "123456789012"  
}  
}  
}  
}
```

Monitoramento e marcação no AWS Elemental MediaStore

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho do AWS Elemental MediaStore e de suas outras AWS soluções. AWS fornece as seguintes ferramentas de monitoramento para observar MediaStore, relatar quando algo está errado e realizar ações automáticas quando apropriado:

- AWS CloudTrail captura chamadas de API e eventos relacionados feitos por ou em nome de sua AWS conta e entrega os arquivos de log para um bucket do Amazon S3 que você especificar. Você pode identificar quais usuários e contas ligaram AWS, o endereço IP de origem a partir do qual as chamadas foram feitas e quando elas ocorreram. Para obter mais informações, consulte o [Guia do usuário do AWS CloudTrail](#).
- A Amazon CloudWatch monitora seus AWS recursos e os aplicativos nos quais você executa AWS em tempo real. Você pode coletar e rastrear métricas, criar painéis personalizados e definir alarmes que o notificam ou que realizam ações quando uma métrica especificada atinge um limite definido. Por exemplo, você pode CloudWatch rastrear o uso da CPU ou outras métricas de suas EC2 instâncias da Amazon e iniciar automaticamente novas instâncias quando necessário. Para obter mais informações, consulte o [Guia CloudWatch do usuário da Amazon](#).
- A Amazon CloudWatch Events fornece um fluxo de eventos do sistema que descrevem mudanças nos AWS recursos. Normalmente, AWS os serviços entregam notificações de CloudWatch eventos aos Eventos em segundos, mas às vezes podem levar um minuto ou mais. CloudWatch Os eventos permitem a computação automatizada baseada em eventos, pois você pode criar regras que observam determinados eventos e acionam ações automatizadas em outros AWS serviços quando esses eventos acontecem. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch Events](#).
- O Amazon CloudWatch Logs permite que você monitore, armazene e acesse seus arquivos de log de EC2 instâncias da Amazon e de outras fontes. CloudTrail CloudWatch Os registros podem monitorar as informações nos arquivos de log e notificá-lo quando determinados limites forem atingidos. É possível também arquivar seus dados de log em armazenamento resiliente. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch Logs](#).

Você também pode atribuir metadados aos seus MediaStore contêineres na forma de tags. Cada tag é um rótulo que consiste em uma chave e um valor definidos por você. As tags podem facilitar o gerenciamento, a pesquisa e a filtragem de recursos. Você pode usar tags para organizar seus AWS

recursos no AWS Management Console, criar relatórios de uso e cobrança em todos os seus AWS recursos e filtrar recursos durante as atividades de automação da infraestrutura.

Tópicos

- [Registro de chamadas da MediaStore API AWS Elemental com AWS CloudTrail](#)
- [Monitoramento do AWS Elemental MediaStore com a Amazon CloudWatch](#)
- [Marcação de recursos do AWS MediaStore Elemental](#)

Registro de chamadas da MediaStore API AWS Elemental com AWS CloudTrail

O AWS Elemental MediaStore é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço em. MediaStore CloudTrail captura um subconjunto de chamadas de API para eventos MediaStore As, incluindo chamadas do MediaStore console e de chamadas de código para a MediaStore API. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para. MediaStore Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita MediaStore, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e muito mais.

Para saber mais CloudTrail, inclusive como configurá-lo e ativá-lo, consulte o [Guia AWS CloudTrail do usuário](#).

Tópicos

- [Informações do AWS Elemental MediaStore em CloudTrail](#)
- [Exemplo: entradas do arquivo de MediaStore log do AWS Elemental](#)

Informações do AWS Elemental MediaStore em CloudTrail

CloudTrail é ativado em sua AWS conta quando você cria a conta. Quando uma atividade de evento suportada ocorre no AWS Elemental MediaStore, essa atividade é registrada em um CloudTrail evento junto com outros eventos de AWS serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes em sua AWS conta. Para obter mais informações, consulte [Visualização de eventos com histórico de CloudTrail eventos](#).

Para um registro contínuo dos eventos em sua AWS conta, incluindo eventos para MediaStore, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando uma trilha é criada no console, a mesma é aplicada a todas as regiões da AWS. A trilha registra eventos de todas as regiões na AWS partição e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para obter mais informações, consulte os tópicos a seguir.

- [Visão geral da criação de uma trilha](#)
- [CloudTrail Serviços e integrações compatíveis](#)
- [Configurando notificações do Amazon SNS para CloudTrail](#)
- [Recebendo arquivos de CloudTrail log de várias regiões](#) e [recebendo arquivos de CloudTrail log de várias contas](#)

O AWS Elemental MediaStore oferece suporte ao registro das seguintes operações como eventos em arquivos de CloudTrail log:

- [CreateContainer](#)
- [DeleteContainer](#)
- [DeleteContainerPolicy](#)
- [DeleteCorsPolicy](#)
- [DescribeContainer](#)
- [GetContainerPolicy](#)
- [GetCorsPolicy](#)
- [ListContainers](#)
- [PutContainerPolicy](#)
- [PutCorsPolicy](#)

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar o seguinte:

- Se a solicitação foi feita com credenciais de usuário raiz ou credenciais de usuário
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado

- Se a solicitação foi feita por outro AWS serviço

Para obter mais informações, consulte [Elemento userIdentity do CloudTrail](#).

Exemplo: entradas do arquivo de MediaStore log do AWS Elemental

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer origem e inclui informações sobre a ação solicitada, a data e hora da ação, parâmetros de solicitação, e assim por diante. arquivos de log do CloudTrail não são um rastreamento de pilha ordenada das chamadas da API pública. Assim, elas não são exibidas em nenhuma ordem específica.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a CreateContainer operação:

```
{
  "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":
    "MNCTGH4HRQJ27GRMBVDPIVHEP4L02BN6MUVHBCPSHOAWNSOKSXC024B2UE0BBND5D0NRXTMFK3TOJ4G7AHWMESI",
    "eventID": "7085b140-fb2c-409b-a329-f567912d704c",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }
}
```

Monitoramento do AWS Elemental MediaStore com a Amazon CloudWatch

Você pode monitorar o AWS Elemental MediaStore usando CloudWatch, que coleta dados brutos e os processa em métricas legíveis. CloudWatch mantém estatísticas por 15 meses para que você possa acessar informações históricas e ter uma melhor perspectiva sobre o desempenho de seu aplicativo ou serviço web. Você também pode definir alarmes que observam determinados limites e enviam notificações ou realizam ações quando esses limites são atingidos. Para obter mais informações, consulte o [Guia CloudWatch do usuário da Amazon](#).

AWS fornece as seguintes ferramentas de monitoramento para observar MediaStore, relatar quando algo está errado e realizar ações automáticas quando apropriado:

- O Amazon CloudWatch Logs permite que você monitore, armazene e acesse seus arquivos de log a partir de AWS serviços como o AWS Elemental MediaStore. Você pode usar o CloudWatch Logs para monitorar aplicativos e sistemas usando dados de log. Por exemplo, o CloudWatch Logs pode rastrear o número de erros que ocorrem nos registros do seu aplicativo e enviar uma notificação sempre que a taxa de erros exceder um limite especificado por você. CloudWatch O Logs usa seus dados de registro para monitoramento, portanto, nenhuma alteração no código é necessária. Por exemplo, você pode monitorar registros de aplicativos em busca de termos literais específicos (como "ValidationException") ou contar o número de PutObject solicitações feitas durante um determinado período de tempo. Quando o termo que você está procurando é

encontrado, o CloudWatch Logs reporta os dados para uma CloudWatch métrica especificada por você. Os dados de log são criptografados em trânsito e em repouso.

- A Amazon CloudWatch Events oferece eventos do sistema que descrevem mudanças nos AWS recursos, como MediaStore objetos. Normalmente, AWS os serviços entregam notificações de CloudWatch eventos aos Eventos em segundos, mas às vezes podem levar um minuto ou mais. Você pode configurar regras para combinar eventos (como uma `DeleteObject` solicitação) e roteá-los para uma ou mais funções ou fluxos de destino. CloudWatch Os eventos ficam cientes das mudanças operacionais à medida que elas ocorrem. Além disso, o CloudWatch Events responde a essas mudanças operacionais e toma medidas corretivas conforme necessário, enviando mensagens para responder ao ambiente, ativando funções, fazendo alterações e capturando informações de estado.

CloudWatch Registros

O registro de acesso em log fornece detalhes sobre as solicitações que são feitas aos objetos de um contêiner. Os logs de acesso são úteis para muitas aplicações, como auditorias de segurança e acesso. Eles também podem ajudar você a conhecer sua base de clientes e entender sua MediaStore fatura. CloudWatch Os registros são categorizados da seguinte forma:

- Fluxo de logs é uma sequência de eventos de log que compartilham a mesma origem.
- Um grupo logs é um grupo de fluxos de log que compartilham as mesmas configurações de retenção, monitoramento e controle de acesso. Quando você ativa o registro de acesso em um contêiner, MediaStore cria um grupo de registros com um nome como `/aws/mediastore/MyContainerName`. Você pode definir grupos de logs e especificar quais fluxos colocar em cada grupo. Não há cota para o número de streams de log que podem pertencer a um grupo de logs.

Por padrão, os logs são mantidos indefinidamente e nunca expiram. Você pode ajustar a política de retenção para cada grupo de logs, mantendo a retenção indefinida ou escolhendo um período de retenção de um dia a 10 anos.

Configurando permissões para a Amazon CloudWatch

Use AWS Identity and Access Management (IAM) para criar uma função que dê ao AWS Elemental MediaStore acesso à Amazon. CloudWatch Você deve executar essas etapas para que CloudWatch os registros sejam publicados em sua conta. CloudWatch publica automaticamente métricas para sua conta.

Para permitir o MediaStore acesso a CloudWatch

1. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação do console do IAM, escolha Políticas (Políticas) e Create policy (Criar política).
3. Selecione a guia JSON e cole a política a seguir:

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

Essa política permite MediaStore criar grupos e fluxos de registros para qualquer contêiner em qualquer região da sua AWS conta.

4. Selecione Revisar política.
5. Na página Review policy (Revisar política), em Name (Nome), digite **MediaStoreAccessLogsPolicy** e escolha Create policy (Criar política).
6. No painel de navegação do console do IAM, escolha Perfis e, em seguida, Criar perfil.
7. Escolha o tipo de função Another AWS account (Outra conta da AWS).
8. Em ID da conta, insira o ID AWS da sua conta.
9. Escolha Próximo: Permissões.

10. Na caixa de pesquisa, insira **MediaStoreAccessLogsPolicy**.
11. Marque a caixa de seleção ao lado de sua nova política e escolha Next: Tags (Próximo: Tags).
12. Escolha Next: Review (Próximo: rever) para visualizar o novo usuário.
13. Em Role name (Nome da função), digite **MediaStoreAccessLogs** e escolha Create role (Criar função).
14. Na mensagem de confirmação, escolha o nome da função que você acabou de criar (**MediaStoreAccessLogs**).
15. Na página Summary (Resumo) da função, escolha a guia Trust relationship (Relação de confiança).
16. Selecione Edit trust relationship (Editar relação de confiança).
17. No documento da política, altere a entidade principal para o serviço MediaStore. A aparência deve ser semelhante a esta:

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

Toda a política deve ser lida da seguinte maneira:

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

18. Selecione Atualizar política de confiança.

Habilitar registro em log de acessos para um contêiner

Por padrão, o AWS Elemental MediaStore não coleta registros de acesso. Quando você ativa o registro de acesso em um contêiner, MediaStore entrega os registros de acesso para objetos armazenados nesse contêiner para a Amazon CloudWatch. Os logs de acesso fornecem registros detalhados para solicitações feitas para qualquer objeto armazenado no contêiner. Essa informação pode incluir o tipo de solicitação, os recursos que foram especificados na solicitação e a hora e data em que a solicitação foi processada.

Important

Não há custo adicional para a habilitação do registro de acesso em um contêiner do MediaStore. No entanto, todos os arquivos de log que o serviço fornece acumulam as cobranças normais de armazenamento. (Você pode excluir os arquivos de log a qualquer momento.) A AWS não avalia cobranças de transferência de dados para fornecimento de arquivos de log, mas cobra a taxa de transferência de dados normal para acessar os arquivos de log.

Para habilitar o registro de acesso em logs (AWS CLI)

- No AWS CLI, use o `start-access-logging` comando:

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

Este comando não tem valor de retorno.

Desabilitar o registro em log de acessos para um contêiner

Quando você desativa o registro de acesso em um contêiner, o AWS Elemental MediaStore para de enviar registros de acesso para a Amazon CloudWatch. Esses logs de acesso não são salvos e não são recuperáveis.

Para desabilitar o registro de acesso em log (AWS CLI)

- No AWS CLI, use o `stop-access-logging` comando:

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

Este comando não tem valor de retorno.

Solução de problemas de registro de acesso no AWS Elemental MediaStore

Quando os registros de MediaStore acesso do AWS Elemental não aparecerem na Amazon CloudWatch, consulte a tabela a seguir para ver possíveis causas e soluções.

Note

Certifique-se de ativar AWS CloudTrail os registros para ajudar no processo de solução de problemas.

Sintomas	O problema pode ser...	Experimente Isso...
Você não vê nenhum CloudTrail evento, mesmo que CloudTrail os registros estejam ativados.	A perfil do IAM não existe ou o nome, as permissões ou a política de confiança estão incorretos.	Crie uma função com o nome, permissões e política de confiança corretos. Consulte the section called “Configurando permissões para CloudWatch” .
Você enviou uma solicitação de API <code>DescribeContainer</code> , mas a resposta mostra que o parâmetro <code>AccessLoggingEnabled</code> tem um valor de <code>False</code> . Além disso, você não consegue visualizar nenhum evento do CloudTrail para a função <code>MediaStoreAccessLogs</code> fazendo uma chamada bem-sucedida de <code>DescribeLogGroup</code> , <code>CreateLogGroup</code> , <code>DescribeLogStream</code> ou <code>CreateLogStream</code> .	A perfil do IAM não existe ou o nome, as permissões ou a política de confiança estão incorretos. O registro de acesso em logs não está ativado no contêiner.	Crie uma função com o nome, permissões e política de confiança corretos. Consulte the section called “Configurando permissões para CloudWatch” . Habilitar logs de acesso para o contêiner. Consulte the section called “Habilitar registro em log de acessos” .

Sintomas	O problema pode ser...	Experimente Isso...
<p>No CloudTrail console, você vê um evento com um erro de acesso negado relacionado à MediaStoreAccessLogs função. O CloudTrail evento pode incluir linhas como as seguintes:</p> <pre>"eventSource": "logs.amazonaws.com", "errorCode": "AccessDenied", "errorMessage": "User: arn:aws:sts::11112223333:assumed-role/MediaStoreAccessLogs/MediaStoreAccessLogsSession is not authorized to perform: logs:DescribeLogGroups on resource: arn:aws:logs:us-west-2:11112223333:log-group::log-stream:",</pre>	<p>A função do IAM não tem as permissões corretas para o AWS Elemental MediaStore.</p>	<p>Atualize o perfil do IAM para ter as permissões e política de confiança corretas. Consulte the section called "Configurando permissões para CloudWatch".</p>
<p>Você não vê nenhum log para um contêiner inteiro ou contêineres.</p>	<p>Sua conta pode ter excedido a CloudWatch cota de grupos de registros por conta por região. Veja as cotas para grupos de registros no Guia do usuário do Amazon CloudWatch Logs.</p>	<p>No CloudWatch console, determine se sua conta atingiu a CloudWatch cota para grupos de registros. Se necessário, solicite um aumento da cota.</p>

Sintomas	O problema pode ser...	Experimente Isso...
Você vê alguns registros CloudWatch, mas não todos os registros que você espera ver.	Sua conta pode ter excedido a CloudWatch cota de transações por segundo por conta por região. Veja as cotas PutLogEvents no Guia do usuário do Amazon CloudWatch Logs .	Solicite um aumento de cota para CloudWatch transações por segundo por conta por região.

Formato de log de acessos

Os arquivos de log de acesso consistem em uma sequência de registros de log em formato JSON, em que cada registro de log representa uma solicitação. A ordem dos campos dentro do log pode variar. Veja a seguir um exemplo de log que consiste em dois registros de log:

```
{
  "Path": "/FootballMatch/West",
  "Requester": "arn:aws:iam::111122223333:user/maria-garcia",
  "AWSAccountId": "111122223333",
  "RequestID":
"aaaAAA111bbbBBB222cccCCC333dddDDD444eeeEEE555ffffFFF666gggGGG777hhhHHH888iiiIII999jjjJJJ",
  "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":
"dddDDD444eeeEEE555ffffFFF666gggGGG777hhhHHH888iiiIII999jjjJJJ000cccCCC333bbbBBB222aaaAAA",
"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
}

```

A lista a seguir descreve os campos dos registros de log:

AWSAccountIdentificação

O AWS ID da conta que foi usada para fazer a solicitação.

BytesReceived

O número de bytes no corpo da solicitação recebidos pelo servidor do MediaStore.

BytesSent

O número de bytes no corpo da resposta enviados pelo servidor do MediaStore. Esse valor geralmente é o mesmo que o valor do cabeçalho Content-Length incluído com respostas do servidor.

ContainerName

O nome do contêiner que recebeu a solicitação.

ErrorCode

O código de MediaStore erro (como `InternalServerError`). Se nenhum erro tiver ocorrido, o caractere - será exibido. Um código de erro poderá ser exibido mesmo se o código de status for 200 (o que indica uma conexão fechada ou um erro depois de o servidor iniciar a transmissão da resposta).

ExpiresAt

A data e a hora de expiração do objeto. Esse valor é baseado na idade de expiração definida por [transient data rule](#) na política de ciclo de vida aplicada ao contêiner. O valor é horário e data ISO-8601 baseados no relógio do sistema do host que atendeu a solicitação. Se a política de ciclo de vida não tiver uma regra de dados transitória que se aplique ao objeto, ou se não houver uma política de ciclo de vida aplicada ao contêiner, o valor desse campo será null. Esse campo será aplicado apenas às seguintes operações: PutObject, GetObject, DescribeObject e DeleteObject.

HTTPStatus

O código numérico do status do HTTP da resposta.

Operação

A operação executada, como PutObject ou ListItems.

Path

O caminho dentro do contêiner no qual o objeto está armazenado. Se a operação não leva um parâmetro de caminho, o caractere - é exibido.

ReceivedTime

A hora do dia em que a solicitação foi recebida. O valor é horário e data ISO-8601 baseados no relógio do sistema do host que atendeu a solicitação.

Solicitante

O usuário Nome de recurso da Amazon (ARN) da conta que foi usada para fazer a solicitação. Para solicitações não autenticadas, esse valor será anonymous. Se a solicitação falhar antes de a autenticação ser concluída, esse campo poderá estar ausente do log. Para tais solicitações, o ErrorCode pode identificar o problema de autorização.

RequestID

Uma string gerada pelo AWS Elemental MediaStore para identificar de forma exclusiva cada solicitação.

Origem

O endereço da Internet aparente do solicitante ou do gerente do serviço da AWS que está fazendo a chamada. Se os firewalls e proxies intermediários obscurecerem o endereço da máquina que fez a solicitação, o valor será definido como nulo.

TotalTime

O número de milissegundos (ms) em que a solicitação esteve em andamento da perspectiva do servidor. Esse valor é medido a partir do momento em que o pedido é recebido pelo serviço e termina no momento em que o último byte da resposta é enviado. Esse valor é medido do ponto de vista do servidor porque medições feitas da perspectiva do cliente são afetadas pela latência de rede.

TurnAroundTime

O número de milissegundos MediaStore gastos no processamento da sua solicitação. Esse valor é medido do momento do recebimento do último byte da solicitação até o momento em que o primeiro byte da resposta foi enviado.

A ordem dos campos no log pode variar.

As alterações do status dos registros em log entram em vigor ao longo do tempo

As alterações no status do registro em log de um contêiner levam tempo para realmente afetar a entrega de arquivos de log. Por exemplo, se você habilitar o registro em log para um contêiner A, algumas solicitações feitas na hora seguinte podem ser registradas, enquanto outras não. Se você desabilitar o registro em log para o contêiner B, alguns logs poderão continuar a ser entregues na hora seguinte, enquanto outros não. Em todos os casos, as novas configurações acabam sendo aplicadas sem que você necessite tomar qualquer outra ação.

Entrega de logs do servidor de melhor esforço

Os registros de log de acessos são entregues com base no melhor esforço. A maioria das solicitações para um contêiner configurado corretamente para registro em log tem como resultado um registro do log entregue. A maioria dos registros de log é entregue dentro de algumas horas após o tempo em que forem registrados, mas eles podem ser entregues com mais frequência.

A integralidade e a pontualidade do registro de acesso em log não são garantidas. O registro de log de uma solicitação específica pode ser entregue muito depois de a solicitação ter sido realmente processada ou pode nem ser entregue. A finalidade dos logs de acesso é proporcionar uma ideia da natureza do tráfego no contêiner. É raro perder registros de log, mas o registro de acesso em logs não tem como objetivo ser uma contabilidade completa de todas as solicitações.

Levando em conta a natureza de melhor esforço do atributo de registro de acesso em log, os relatórios de uso disponíveis no portal da AWS (relatórios do Gerenciamento de custos e faturamento

no [AWS Management Console](#)) podem incluir uma ou mais solicitações de acesso que não aparecem em um log de acesso entregue.

Considerações de programação para o formato de logs de acesso

Periodicamente, você pode estender o formato de log de acesso adicionando novos campos. O código que analisa os logs de acesso devem ser escritos para tratar de campos adicionais que não compreende.

CloudWatch Eventos

O Amazon CloudWatch Events permite que você automatize seus AWS serviços e responda automaticamente a eventos do sistema, como problemas de disponibilidade de aplicativos ou alterações de recursos. Você pode escrever regras simples para indicar quais eventos são do seu interesse, e as ações automatizadas a serem tomadas quando um evento corresponder à regra.

Important

Normalmente, AWS os serviços entregam notificações de CloudWatch eventos aos Eventos em segundos, mas às vezes podem levar um minuto ou mais.

Quando um arquivo é carregado em um contêiner ou removido de um contêiner, dois eventos são acionados em sucessão no CloudWatch serviço:

1. [the section called “Evento de alteração de estado do objeto”](#)
2. [the section called “Evento de alteração de estado do contêiner”](#)

Para obter informações sobre como se inscrever nesses eventos, consulte [Amazon CloudWatch](#).

Ações que podem ser automaticamente acionadas incluem:

- Invocando uma função AWS Lambda
- Invocando o EC2 comando Amazon Run
- Transmitir o evento Amazon Kinesis Data Streams
- Ativando uma máquina de AWS Step Functions estado
- Notificar um tópico ou uma fila do Amazon SNS AWS SMS

Alguns exemplos de uso de CloudWatch eventos com o AWS Elemental MediaStore incluem o seguinte:

- Ativar uma função do Lambda sempre que um contêiner é criado
- Notificar um tópico do Amazon SNS quando um objeto é excluído

Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch Events](#).

Tópicos

- [Evento de mudança de estado do MediaStore objeto AWS Elemental](#)
- [Evento de mudança de estado do MediaStore contêiner AWS Elemental](#)

Evento de mudança de estado do MediaStore objeto AWS Elemental

Esse evento é publicado quando o estado de um objeto for alterado, quando o objeto for carregado ou excluído.

Note

Objetos que expiram devido a uma regra de dados transitória não emitem um CloudWatch evento quando expiram.

Para obter informações sobre como se inscrever neste evento, consulte [Amazon CloudWatch](#).

Objeto atualizado

```
{
  "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 removido

```

{
  "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 mudança de estado do MediaStore contêiner AWS Elemental

Esse evento é publicado quando o estado de um contêiner for alterado, quando um contêiner for adicionado ou excluído. Para obter informações sobre como se inscrever neste evento, consulte [Amazon CloudWatch](#).

Contêiner criado

```
{
```

```
"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"
}
}
```

Contêiner removido

```
{
  "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"
  }
}
```

Monitoramento do AWS Elemental com métricas MediaStore da Amazon CloudWatch

Você pode monitorar o AWS Elemental MediaStore usando CloudWatch, que coleta dados brutos e os processa em métricas legíveis. CloudWatch mantém as estatísticas por 15 meses para que você possa acessar informações históricas e ter uma melhor perspectiva sobre o desempenho de

seu aplicativo ou serviço web. Você também pode definir alarmes que observam determinados limites e enviam notificações ou realizam ações quando esses limites são atingidos. Para obter mais informações, consulte o [Guia CloudWatch do usuário da Amazon](#).

Para o AWS Elemental MediaStore, talvez você queira assistir BytesDownloaded e enviar um e-mail para si mesmo quando essa métrica atingir um determinado limite.

Para visualizar métricas usando o CloudWatch console

As métricas são agrupadas primeiro pelo namespace do serviço e, em seguida, por várias combinações de dimensão dentro de cada namespace.

1. Faça login no AWS Management Console e abra o CloudWatch console em <https://console.aws.amazon.com/cloudwatch/>.
2. No painel de navegação, selecione Métricas.
3. Em Todas as métricas, escolha o namespace MediaStore AWS/.
4. Escolha a dimensão métrica para visualizar as métricas. Por exemplo, escolha Request metrics by container para visualizar métricas para os diferentes tipos de solicitações que foram enviadas para o contêiner.

Para visualizar métricas usando o AWS CLI

- Em um prompt de comando, use o seguinte comando:

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

Métricas do AWS Elemental MediaStore

A tabela a seguir lista as métricas para as quais o AWS Elemental MediaStore envia. CloudWatch

Note

Para visualizar as métricas, você deve [adicionar uma política métrica](#) ao contêiner para permitir MediaStore o envio de métricas para a Amazon CloudWatch.

Métrica	Descrição
RequestCount	<p>O número total de solicitações HTTP feitas para um contêiner do MediaStore, separadas por tipo de operação (Put, Get, Delete, Describe, List).</p> <p>Unidades: contagem</p> <p>Dimensões válidas:</p> <ul style="list-style-type: none">• Nome do contêiner• Nome do grupo de objetos• Tipos de solicitação <p>Estatísticas válidas: soma</p>
4xxErrorCount	<p>O número de solicitações HTTP feitas para MediaStore isso resultou em um erro 4xx.</p> <p>Unidades: contagem</p> <p>Dimensões válidas:</p> <ul style="list-style-type: none">• Nome do contêiner• Nome do grupo de objetos• Tipos de solicitação <p>Estatísticas válidas: soma</p>
5xxErrorCount	<p>O número de solicitações HTTP feitas para MediaStore isso resultou em um erro 5xx.</p> <p>Unidades: contagem</p> <p>Dimensões válidas:</p> <ul style="list-style-type: none">• Nome do contêiner• Nome do grupo de objetos• Tipos de solicitação

Métrica	Descrição
	Estatísticas válidas: soma
BytesUploaded	<p>O número de bytes carregados para solicitações feitas a um contêiner do MediaStore , em que a solicitação inclui um corpo.</p> <p>Unidades: bytes</p> <p>Dimensões válidas:</p> <ul style="list-style-type: none"> • Nome do contêiner • Nome do grupo de objetos <p>Estatísticas válidas: média (bytes por solicitação), soma (bytes por período), contagem de amostras, mín. (o mesmo que P0,0), máx. (o mesmo que P100), qualquer percentil entre P0,0 e P99,9</p>
BytesDownloaded	<p>O número de bytes obtidos por download para solicitações feitas a um contêiner do MediaStore , em que a resposta inclui um corpo.</p> <p>Unidades: bytes</p> <p>Dimensões válidas:</p> <ul style="list-style-type: none"> • Nome do contêiner • Nome do grupo de objetos <p>Estatísticas válidas: média (bytes por solicitação), soma (bytes por período), contagem de amostras, mín. (o mesmo que P0,0), máx. (o mesmo que P100), qualquer percentil entre P0,0 e P99,9</p>

Métrica	Descrição
TotalTime	<p>O número de milissegundos (ms) em que a solicitação esteve em trânsito da perspectiva do servidor. Esse valor é medido desde o momento em que MediaStore recebe sua solicitação até o momento em que envia o último byte da resposta. Esse valor é medido do ponto de vista do servidor porque medições feitas da perspectiva do cliente são afetadas pela latência de rede.</p> <p>Unidade: milissegundos</p> <p>Dimensões válidas:</p> <ul style="list-style-type: none">• Nome do contêiner• Nome do grupo de objetos• Tipos de solicitação <p>Estatísticas válidas: média, mín. (igual a P0,0), máx. (o mesmo que P100), qualquer percentil entre P0,0 e P100</p>
TurnaroundTime	<p>O número de milissegundos MediaStore gastos no processamento da sua solicitação. Esse valor é medido desde o momento em que MediaStore recebe o último byte da sua solicitação até o momento em que ele envia o primeiro byte da resposta.</p> <p>Unidade: milissegundos</p> <p>Dimensões válidas:</p> <ul style="list-style-type: none">• Nome do contêiner• Nome do grupo de objetos• Tipos de solicitação <p>Estatísticas válidas: média, mín. (igual a P0,0), máx. (o mesmo que P100), qualquer percentil entre P0,0 e P100</p>

Métrica	Descrição
ThrottledCount	<p>O número de solicitações HTTP feitas para MediaStore isso foi limitado.</p> <p>Unidades: contagem</p> <p>Dimensões válidas:</p> <ul style="list-style-type: none">• Nome do contêiner• Nome do grupo de objetos• Tipos de solicitação <p>Estatísticas válidas: soma</p>

Marcação de recursos do AWS MediaStore Elemental

Uma tag é um rótulo de atributo personalizado que você atribui ou AWS atribui a um AWS recurso. Cada tag tem duas partes:

- Uma chave de tag (por exemplo `CostCenter`, `Environment` ou `Project`). Chaves de tag fazem distinção entre maiúsculas e minúsculas.
- Um campo opcional conhecido como um valor de tag (por exemplo, `111122223333` ou `Production`). Omitir o valor da tag é o mesmo que usar uma string vazia. Como chaves de tag, os valores das tags diferenciam maiúsculas de minúsculas.

As tags ajudam a:

- Identifique e organize seus AWS recursos. Muitos serviços da AWS oferecem suporte à marcação para que você possa atribuir a mesma tag a recursos de diferentes serviços para indicar que os recursos estão relacionados. Por exemplo, você pode atribuir a mesma tag a um AWS Elemental MediaStore *container* que você atribui a uma AWS Elemental MediaLive entrada.
- Monitorar seus custos da AWS. Você ativa essas tags no Gerenciamento de Faturamento e Custos da AWS painel. A AWS usa as tags para categorizar seus custos e fornecer um relatório mensal de alocação de custos mensais a você. Para obter mais informações, consulte [Usar tags de alocação de custos](#) no [Guia do usuário do AWS Billing](#).

As seções a seguir fornecem mais informações sobre tags para o AWS Elemental MediaStore.

Recursos compatíveis no AWS Elemental MediaStore

Os seguintes recursos no AWS Elemental MediaStore oferecem suporte à marcação:

- *container*

Para obter informações sobre como adicionar e gerenciar tags, consulte [Como gerenciar tags](#).

O AWS Elemental MediaStore não oferece suporte ao recurso de controle de acesso baseado em tags do AWS Identity and Access Management (IAM).

Convenções de uso e nomenclatura de tags

As seguintes convenções básicas de nomenclatura e uso se aplicam ao uso de tags com recursos do AWS MediaStore Elemental:

- Cada recurso pode ter um máximo de 50 tags.
- Em todos os recursos, cada chave de tag deve ser exclusiva e pode ter apenas um valor.
- O comprimento máximo da chave da tag é de 128 caracteres Unicode em UTF-8.
- O comprimento máximo do valor da tag é de 256 caracteres Unicode em UTF-8.
- Os caracteres permitidos são letras, números, espaços representáveis em UTF-8, além dos seguintes caracteres: . : + = @ _ / - (hífen). EC2 Os recursos da Amazon permitem qualquer personagem.
- As chaves e valores das tags diferenciam maiúsculas de minúsculas. Como melhor prática, adote uma estratégia para letras maiúsculas em tags e implemente-a de forma consistente em todos os tipos de recursos. Por exemplo, decida se deseja usar Costcenter, costcenter ou CostCenter e use a mesma convenção para todas as tags. Evite usar tags semelhantes com tratamento do tamanho de letra inconsistente.
- O `aws:` prefixo é proibido para tags; está reservado para AWS uso. Você não pode editar nem excluir chaves nem valores de tags com esse prefixo. As tags com esse prefixo não contam para as tags por cota de recurso.

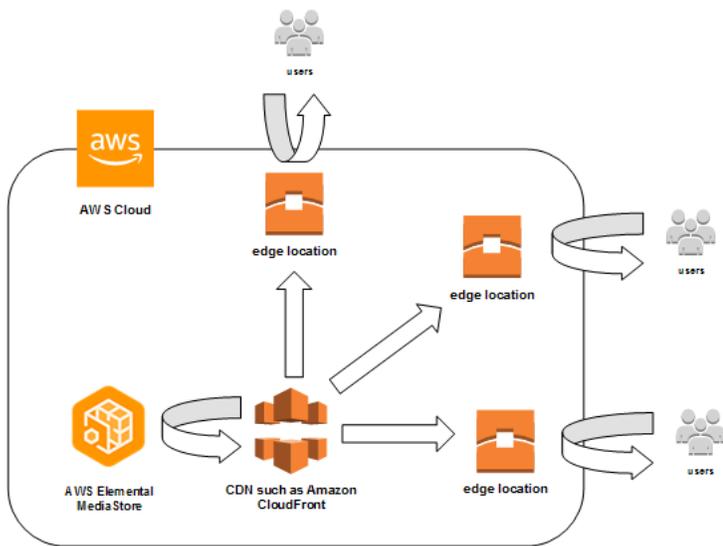
Como gerenciar tags

As tags são compostas de propriedades Value e Key em um recurso. Você pode usar a AWS CLI ou a MediaStore API para adicionar, editar ou excluir os valores dessas propriedades. Para obter informações sobre como trabalhar com tags, consulte as seguintes seções na Referência da MediaStore API Elemental da AWS:

- [CreateContainer](#)
- [ListTagsForResource](#)
- [Recursos](#)
- [TagResource](#)
- [UntagResource](#)

Trabalhando com redes de distribuição de conteúdo (CDNs)

Você pode usar uma rede de entrega de conteúdo (CDN), como CloudFront a [Amazon](#), para servir o conteúdo que você armazena no AWS MediaStore Elemental. Uma CDN é um conjunto globalmente distribuído de servidores que armazena conteúdo em cache como vídeos. Quando um usuário solicita o conteúdo, a CDN encaminha a solicitação para o local da borda que fornece a latência mais baixa. Se o conteúdo já está armazenado em cache nesse local da borda, a CDN o entrega imediatamente. Se seu conteúdo não estiver atualmente nesse ponto de presença, a CDN o recupera de sua origem (como seu MediaStore contêiner) e o distribui para o usuário.



Tópicos

- [Permitindo que CloudFront a Amazon acesse seu contêiner AWS Elemental MediaStore](#)
- [Interação do AWS Elemental MediaStore com caches HTTP](#)

Permitindo que CloudFront a Amazon acesse seu contêiner AWS Elemental MediaStore

Você pode usar CloudFront a Amazon para servir o conteúdo que você armazena em um contêiner no AWS Elemental MediaStore. É possível fazer isso das seguintes maneiras:

- [Uso do controle de acesso de origem \(OAC\)](#)- (Recomendado) Use essa opção se você Região da AWS suportar o recurso OAC do CloudFront.

- [Uso de segredos compartilhados](#)- Use essa opção se você Região da AWS não suportar o recurso OAC do CloudFront.

Uso do controle de acesso de origem (OAC)

Você pode usar o recurso Origin Access Control (OAC) da Amazon CloudFront para proteger as origens do AWS MediaStore Elemental com segurança aprimorada. Você pode habilitar o [AWS Signature Version 4 \(SigV4\)](#) nas CloudFront solicitações de MediaStore origens e definir quando e se CloudFront deve assinar as solicitações. Você pode acessar o recurso OAC CloudFront por meio do console APIs, SDK ou CLI, e não há taxas adicionais por seu uso.

Para obter mais informações sobre como usar o recurso OAC com MediaStore, consulte [Restringir o acesso a uma MediaStore origem](#) no [Amazon CloudFront Developer Guide](#).

Uso de segredos compartilhados

Se você Região da AWS não oferece suporte ao recurso OAC da Amazon CloudFront, você pode anexar uma política ao seu contêiner do AWS MediaStore Elemental que conceda acesso de leitura ou superior a CloudFront

Note

Recomendamos usar o recurso OAC se você o Região da AWS suportar. Os procedimentos a seguir exigem que você configure MediaStore e CloudFront use segredos compartilhados para restringir o acesso aos MediaStore contêineres. Para seguir as práticas recomendadas de segurança, essa configuração manual exige uma rotação periódica de segredos. Com o OAC nas MediaStore origens, você pode CloudFront instruir a assinar solicitações usando o SigV4 e encaminhá-las MediaStore para correspondência de assinaturas, eliminando a necessidade de usar e alternar segredos. Isso garante que as solicitações sejam verificadas automaticamente antes que o conteúdo de mídia seja veiculado, tornando a entrega de conteúdo de mídia mais CloudFront simples MediaStore e segura.

Para permitir CloudFront o acesso ao seu contêiner (console)

1. Abra o MediaStore console em <https://console.aws.amazon.com/mediastore/>.
2. Na página Containers (Contêineres), escolha o nome do contêiner.

A página de detalhes do contêiner é exibida.

3. Na seção Política de contêineres, anexe uma política que conceda acesso de leitura ou superior à Amazon CloudFront.

Example

O exemplo de política a seguir, que é similar ao exemplo de política para [Acesso de leitura pública por HTTPS](#), está de acordo com esses requisitos, pois permite comandos `GetObject` e `DescribeObject` de qualquer pessoa que envia solicitações para seu domínio por meio de HTTPS. Além disso, o exemplo de política a seguir protege melhor seu fluxo de trabalho porque permite CloudFront acesso a MediaStore objetos somente quando a solicitação ocorre por meio de uma conexão HTTPS e contém o cabeçalho `Referer` correto.

```
{
  "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. Na seção Container CORS policy (Política de CORS de contêiner), atribua uma política que permita o nível de acesso apropriado.

Note

Uma [política de CORS](#) é necessária somente se você deseja conceder acesso a um jogador baseado em navegador.

5. Anote os seguintes detalhes:
 - O endpoint de dados que é atribuído a seu contêiner do . Você pode encontrar essas informações na seção Info (Informações) da página Containers (Contêineres). Em CloudFront, o endpoint de dados é chamado de nome de domínio de origem.
 - A estrutura de pastas no contêiner em que os objetos são armazenados. Em CloudFront, isso é chamado de caminho de origem. Essa configuração é opcional. Para obter mais informações sobre caminhos de origem, consulte o [Amazon CloudFront Developer Guide](#).
6. Em CloudFront, crie uma distribuição [configurada para servir conteúdo do AWS Elemental MediaStore](#). Você precisará das informações coletadas na etapa anterior.

Depois de anexar a política aos seus MediaStore contêineres, você deve configurar CloudFront para usar somente conexões HTTPS para solicitações de origem e também adicionar um cabeçalho personalizado com o valor secreto correto.

Para configurar CloudFront para acessar seu contêiner por meio de uma conexão HTTPS com um valor secreto para o cabeçalho Referer (console)

1. Abra o CloudFront console.
2. Na página Origins, escolha sua MediaStore origem.
3. Escolha Editar.
4. Selecione Somente HTTPS para o protocolo.
5. Na seção Adicionar cabeçalho personalizado, escolha Adicionar cabeçalho.
6. Para Nome, escolha Referer. Para o valor, use a mesma `<secretValue>` string que você usou na sua política de contêiner.
7. Escolha Salvar e deixe as alterações serem implantadas.

Interação do AWS Elemental MediaStore com caches HTTP

O AWS Elemental MediaStore armazena objetos para que eles possam ser armazenados em cache de forma correta e eficiente por redes de distribuição de conteúdo () CDNs como a Amazon. CloudFront Quando um usuário final ou CDN recupera um objeto MediaStore, o serviço retorna cabeçalhos HTTP que afetam o comportamento de armazenamento em cache do objeto. (Os padrões para o comportamento de armazenamento em cache do HTTP 1.1 são encontrados na [RFC2616 seção 13.](#)) Estes cabeçalhos são:

- **ETag** (não personalizável) – o cabeçalho da tag de entidade é um identificador exclusivo para a resposta enviada pelo MediaStore . Os navegadores da Web CDNs e compatíveis com os padrões usam essa tag como uma chave para armazenar o objeto em cache. MediaStore gera automaticamente um ETag para cada objeto quando ele é carregado. Você pode [ver os detalhes de um objeto](#) para determinar seu ETag valor.
- **Last-Modified**(não personalizável) — O valor desse cabeçalho indica a data e a hora em que o objeto foi modificado. MediaStore gera automaticamente esse valor quando o objeto é carregado.
- **Cache-Control** (personalizável) – o valor deste cabeçalho controla por quanto tempo um objeto deve permanecer armazenado em cache antes de a CDN verificar se ele foi modificado. [Você pode definir esse cabeçalho com qualquer valor ao carregar um objeto em um MediaStore contêiner usando a CLI ou a API.](#) O conjunto completo de valores válidos é descrito na [documentação HTTP/1.1](#). Se você não definir esse valor ao carregar um objeto, MediaStore não retornará esse cabeçalho quando o objeto for recuperado.

Um caso de uso comum para o cabeçalho Cache-Control é a especificação de uma duração para armazenar o objeto em cache. Por exemplo, suponha que você tenha um arquivo de manifesto de vídeo que está sendo frequentemente substituído por um codificador. É possível definir a max-age como 10 para indicar que o objeto deve permanecer armazenado em cache por apenas 10 segundos. Ou suponha que você tenha um segmento de vídeo armazenado que nunca será substituído. É possível definir a max-age para esse objeto como 31536000 para armazená-lo em cache por cerca de 1 ano.

Solicitações condicionais

Solicitações condicionais para MediaStore

MediaStore responde de forma idêntica às solicitações condicionais (usando cabeçalhos de solicitação como If-Modified-Since e If-None-Match, conforme descrito em [RFC7232](#)) e às

solicitações incondicionais. Isso significa que, ao MediaStore receber uma `GetObject` solicitação válida, o serviço sempre retorna o objeto, mesmo que o cliente já tenha o objeto.

Solicitações condicionais para CDNs

CDNs que veiculam conteúdo em nome de MediaStore podem processar solicitações condicionais retornando `304 Not Modified`, conforme descrito na [RFC7232 seção 4.1](#). Isso indica que não há necessidade de transferir o conteúdo completo do objeto, porque o solicitante já tem um objeto correspondente à solicitação condicional.

CDNs (e outros caches compatíveis com HTTP/1.1) baseiam essas decisões nos `Cache-Control` cabeçalhos `ETag` e que são encaminhados pelos servidores de origem. Para controlar a frequência com que os servidores de MediaStore origem CDNs consultam atualizações de objetos recuperados repetidamente, defina `Cache-Control` os cabeçalhos desses objetos ao enviá-los para MediaStore

Usando esse serviço com um AWS SDK

AWS kits de desenvolvimento de software (SDKs) estão disponíveis para muitas linguagens de programação populares. Cada SDK fornece uma API, exemplos de código e documentação que permitem que os desenvolvedores criem facilmente aplicações em seu idioma de preferência.

Documentação do SDK	Exemplos de código
AWS SDK para C++	AWS SDK para C++ exemplos de código
AWS CLI	AWS CLI exemplos de código
AWS SDK para Go	AWS SDK para Go exemplos de código
AWS SDK para Java	AWS SDK para Java exemplos de código
AWS SDK para JavaScript	AWS SDK para JavaScript exemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin exemplos de código
AWS SDK para .NET	AWS SDK para .NET exemplos de código
AWS SDK para PHP	AWS SDK para PHP exemplos de código
Ferramentas da AWS para PowerShell	Ferramentas da AWS para PowerShell exemplos de código
AWS SDK para Python (Boto3)	AWS SDK para Python (Boto3) exemplos de código
AWS SDK para Ruby	AWS SDK para Ruby exemplos de código
AWS SDK para Rust	AWS SDK para Rust exemplos de código
SDK da AWS para SAP ABAP	SDK da AWS para SAP ABAP exemplos de código
AWS SDK for Swift	AWS SDK for Swift exemplos de código

Para obter exemplos específicos deste serviço, consulte [Exemplos de código para MediaStore usar AWS SDKs](#).

 Exemplo de disponibilidade

Não consegue encontrar o que precisa? Solicite um exemplo de código usando o link [Fornecer feedback](#) na parte inferior desta página.

Exemplos de código para MediaStore usar AWS SDKs

Os exemplos de código a seguir mostram como usar MediaStore com um kit AWS de desenvolvimento de software (SDK).

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar perfis de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de código

- [Exemplos básicos de MediaStore uso AWS SDKs](#)
 - [Ações para MediaStore usar AWS SDKs](#)
 - [Use CreateContainer com um AWS SDK ou CLI](#)
 - [Use DeleteContainer com um AWS SDK ou CLI](#)
 - [Use DeleteObject com um AWS SDK](#)
 - [Use DescribeContainer com um AWS SDK ou CLI](#)
 - [Use GetObject com um AWS SDK ou CLI](#)
 - [Use ListContainers com um AWS SDK ou CLI](#)
 - [Use PutObject com um AWS SDK ou CLI](#)

Exemplos básicos de MediaStore uso AWS SDKs

Os exemplos de código a seguir mostram como usar o básico do AWS Elemental MediaStore with AWS SDKs.

Exemplos

- [Ações para MediaStore usar AWS SDKs](#)
 - [Use CreateContainer com um AWS SDK ou CLI](#)
 - [Use DeleteContainer com um AWS SDK ou CLI](#)
 - [Use DeleteObject com um AWS SDK](#)

- [Use DescribeContainer com um AWS SDK ou CLI](#)
- [Use GetObject com um AWS SDK ou CLI](#)
- [Use ListContainers com um AWS SDK ou CLI](#)
- [Use PutObject com um AWS SDK ou CLI](#)

Ações para MediaStore usar AWS SDKs

Os exemplos de código a seguir demonstram como realizar MediaStore ações individuais com AWS SDKs. Cada exemplo inclui um link para GitHub, onde você pode encontrar instruções para configurar e executar o código.

Os exemplos a seguir incluem apenas as ações mais utilizadas. Para obter uma lista completa, consulte a [Referência de APIs do AWS Elemental MediaStore](#).

Exemplos

- [Use CreateContainer com um AWS SDK ou CLI](#)
- [Use DeleteContainer com um AWS SDK ou CLI](#)
- [Use DeleteObject com um AWS SDK](#)
- [Use DescribeContainer com um AWS SDK ou CLI](#)
- [Use GetObject com um AWS SDK ou CLI](#)
- [Use ListContainers com um AWS SDK ou CLI](#)
- [Use PutObject com um AWS SDK ou CLI](#)

Use **CreateContainer** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o CreateContainer.

CLI

AWS CLI

Para criar um contêiner

O exemplo `create-container` a seguir cria um contêiner novo e vazio.

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

Saída:

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

Para obter mais informações, consulte [Criação de um contêiner](#) no Guia do MediaStore usuário do AWS Elemental.

- Para obter detalhes da API, consulte [CreateContainer](#) em Referência de AWS CLI Comandos.

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

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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 obter detalhes da API, consulte [CreateContainer](#) a Referência AWS SDK for Java 2.x da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **DeleteContainer** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o DeleteContainer.

CLI

AWS CLI

Para excluir um contêiner

O exemplo `delete-container` a seguir exclui o contêiner especificado. Você só pode excluir um contêiner se ele não tiver objetos.

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

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo um contêiner no Guia](#) do MediaStore usuário do AWS Elemental.

- Para obter detalhes da API, consulte [DeleteContainer](#) em Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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 obter detalhes da API, consulte [DeleteContainer](#) a Referência AWS SDK for Java 2.x da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **DeleteObject** com um AWS SDK

O código de exemplo a seguir mostra como usar `DeleteObject`.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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 obter detalhes da API, consulte [DeleteObject](#) a Referência AWS SDK for Java 2.x da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **DescribeContainer** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DescribeContainer`.

CLI

AWS CLI

Como visualizar os detalhes de um contêiner

O exemplo `describe-container` a seguir exibe os detalhes do contêiner especificado.

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

Saída:

```
{  
  "Container": {  
    "CreationTime": 1563558086,  
    "AccessLoggingEnabled": false,  
    "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/  
ExampleContainer",  
    "Status": "ACTIVE",  
    "Name": "ExampleContainer",  
    "Endpoint": "https://aaabbbccdddee.data.mediastore.us-  
west-2.amazonaws.com"  
  }  
}
```

Para obter mais informações, consulte [Visualizando os detalhes de um contêiner](#) no Guia do MediaStore usuário do AWS Elemental.

- Para obter detalhes da API, consulte [DescribeContainer](#) em Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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 obter detalhes da API, consulte [DescribeContainer](#) a Referência AWS SDK for Java 2.x da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **GetObject** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `GetObject`.

CLI

AWS CLI

Como fazer download de um objeto

O exemplo `get-object` a seguir faz o download de um objeto no endpoint especificado.

```

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

```


Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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 obter detalhes da API, consulte [GetObjecta](#) Referência AWS SDK for Java 2.x da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **ListContainers** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `ListContainers`.

CLI

AWS CLI

Como visualizar uma lista de contêineres

O exemplo `list-containers` a seguir da lista todos os contêineres associados à sua conta.

```
aws mediastore list-containers
```

Saída:

```
{
  "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 obter mais informações, consulte [Visualizando uma lista de contêineres](#) no Guia do MediaStore usuário do AWS Elemental.

- Para obter detalhes da API, consulte [ListContainers](#) em Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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 obter detalhes da API, consulte [ListContainers](#) na Referência AWS SDK for Java 2.x da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **PutObject** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o PutObject.

CLI

AWS CLI

Como fazer upload de um objeto

O exemplo `put-object` a seguir faz upload de um objeto no contêiner especificado. É possível especificar um caminho de pasta no qual o objeto será salvo dentro do contêiner. Se a pasta já existir, o AWS Elemental MediaStore armazena o objeto na pasta. Se a pasta não existir, o serviço a cria e, em seguida, armazena o objeto na pasta.

```
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
```

Saída:

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

Para obter mais informações, consulte [Carregando um objeto no Guia](#) do MediaStore usuário do AWS Elemental.

- Para obter detalhes da API, consulte [PutObject](#) em Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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 obter detalhes da API, consulte [PutObject](#) Referência AWS SDK for Java 2.x da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Cotas no AWS Elemental MediaStore

O console Service Quotas fornece informações sobre as cotas do AWS MediaStore Elemental. Além de visualizar as cotas padrão, você pode usar o console de Service Quotas para [solicitar aumentos de cota](#) para aquelas que podem ser ajustadas.

A tabela a seguir descreve as cotas, anteriormente chamadas de limites, no AWS Elemental MediaStore. As cotas são o número máximo de recursos ou operações de serviço para sua conta da AWS.

Note

Para atribuir cotas a contêineres individuais em sua conta, entre em contato com o AWS Support ou com o gerente da conta. Essa opção pode ajudar a dividir os limites no nível da conta entre seus contêineres para evitar que um contêiner use toda a sua cota.

Recurso ou operação	Cota padrão	Comentários
Contêineres	100	O número máximo de contêineres que podem ser criados nesta conta.
Níveis de pasta	10	O número máximo de níveis de pasta que podem ser criados em um contêiner. É possível criar quantas pastas quiser, desde que não estejam aninhadas mais de 10 níveis em um contêiner.
Pastas	Ilimitado	É possível criar quantas pastas quiser, desde que não estejam aninhadas mais de 10 níveis em um contêiner.
Tamanho do objeto	25 MB	O tamanho de arquivo máximo de um único objeto.
Objetos	Ilimitado	Você pode fazer upload de quantos objetos quiser para uma pasta ou um contêiner em sua conta.

Recurso ou operação	Cota padrão	Comentários
Taxa de solicitações de API DeleteObject	100	O número máximo de solicitações de operação que você pode fazer por segundo. Solicitações adicionais são limitadas. É possível solicitar um aumento da cota .
Taxa de solicitações de API DescribeObject	1.000	O número máximo de solicitações de operação que você pode fazer por segundo. Solicitações adicionais são limitadas. É possível solicitar um aumento da cota .
Taxa de solicitações de GetObject API para disponibilidade de upload padrão	1.000	O número máximo de solicitações de operação que você pode fazer por segundo. Solicitações adicionais são limitadas. É possível solicitar um aumento da cota .
Taxa de solicitações de GetObject API para disponibilidade de upload de streaming	25	O número máximo de solicitações de operação que você pode fazer por segundo. Solicitações adicionais são limitadas. É possível solicitar um aumento da cota .
Taxa de solicitações de API ListItems	5	O número máximo de solicitações de operação que você pode fazer por segundo. Solicitações adicionais são limitadas. É possível solicitar um aumento da cota .

Recurso ou operação	Cota padrão	Comentários
Taxa de solicitações de PutObject API para codificação de transferência em partes (também conhecida como disponibilidade de upload de streaming)	10	<p>O número máximo de solicitações de operação que você pode fazer por segundo. Solicitações adicionais são limitadas.</p> <p>É possível solicitar um aumento da cota. Na solicitação, especifique o TPS solicitado e o tamanho médio do objeto.</p>
Taxa de solicitações de PutObject API para disponibilidade de upload padrão	100	<p>O número máximo de solicitações de operação que você pode fazer por segundo. Solicitações adicionais são limitadas.</p> <p>É possível solicitar um aumento da cota. Na solicitação, especifique o TPS solicitado e o tamanho médio do objeto.</p>
Regras em uma política de métrica	10	O número máximo de regras que podem ser incluídas em uma política de métrica.
Regras em uma política de ciclo de vida de objetos	10	O número máximo de regras que você pode incluir em uma política de ciclo de vida de objetos.

Informações relacionadas ao AWS Elemental MediaStore

A tabela a seguir lista recursos relacionados que você achará úteis ao trabalhar com o AWS Elemental MediaStore.

- [Aulas e workshops](#) — Links para cursos especializados e baseados em funções, além de laboratórios individualizados para ajudar a aprimorar suas AWS habilidades e ganhar experiência prática.
- [AWS Centro do desenvolvedor](#) — explore tutoriais, baixe ferramentas e saiba mais sobre eventos para AWS desenvolvedores.
- [AWS Ferramentas do desenvolvedor](#) — Links para ferramentas do desenvolvedor SDKs, kits de ferramentas do IDE e ferramentas de linha de comando para desenvolver e gerenciar AWS aplicativos.
- [Centro de recursos de introdução](#) — Saiba como configurar seu aplicativo Conta da AWS, participar da AWS comunidade e lançar seu primeiro aplicativo.
- [Tutoriais práticos — Siga os tutoriais](#) para iniciar seu step-by-step primeiro aplicativo no. AWS
- [AWS Whitepapers](#) — Links para uma lista abrangente de AWS white papers técnicos, abrangendo tópicos como arquitetura, segurança e economia e criados por arquitetos de AWS soluções ou outros especialistas técnicos.
- [AWS Support Center](#) — O hub para criar e gerenciar seus AWS Support casos. Também inclui links para outros recursos úteis, como fóruns, informações técnicas FAQs, status de integridade do serviço AWS Trusted Advisor e.
- [Suporte](#) — A principal página da web com informações sobre Suporte um one-on-one canal de suporte de resposta rápida para ajudá-lo a criar e executar aplicativos na nuvem.
- [Entrar em contato](#): um ponto central de contato para consultas relativas a faturas da AWS , contas, eventos, uso abusivo e outros problemas.
- [AWS Termos do site](#) — Informações detalhadas sobre nossos direitos autorais e nossa marca registrada; sua conta, licença e acesso ao site; e outros tópicos.

Histórico do documento para o guia do usuário

A tabela a seguir descreve a documentação dessa versão do AWS Elemental MediaStore. Para receber notificações sobre atualizações dessa documentação, é possível inscrever-se em um feed RSS.

Alteração	Descrição	Data
Aviso de fim do suporte	Aviso de fim do suporte: em 13 de novembro de 2025, o suporte para o AWS MediaStore Elemental AWS será interrompido. Depois de 13 de novembro de 2025, você não poderá mais acessar o MediaStore console ou MediaStore os recursos. Veja esta postagem em blog para obter mais informações.	12 de novembro de 2024
Melhoria no controle de acesso de origem (OAC)	Foram adicionadas informações sobre como usar o OAC com o AWS MediaStore Elemental.	17 de abril de 2023
Atualizações de cotas	Valor corrigido da cota e descrição para Rules in a Metric Policy.	25 de outubro de 2022
ExpiresAt Campo	Os registros de acesso agora incluem um campo ExpiresAt que indica a data e a hora de expiração do objeto com base nas regras de dados transitórios na política de ciclo de vida do contêiner.	16 de julho de 2020

Regras de transição do ciclo de vida	Agora é possível adicionar uma regra de transição do ciclo de vida à sua política de ciclo de vida do objeto que define objetos a serem movidos para a classe de armazenamento de acesso infrequente (IA) depois que eles atingem uma determinada idade.	20 de abril de 2020
Contêiner vazio	Agora você pode excluir todos os objetos dentro de um contêiner de uma vez.	7 de abril de 2020
Support para CloudWatch métricas da Amazon	Você pode definir uma política métrica para ditar para quais métricas serão MediaStore enviadas CloudWatch.	30 de março de 2020
Curingas em regras de exclusão de objetos	Em uma política de ciclo de vida de objeto, agora é possível usar um curinga em uma regra de exclusão de objetos. Isso permite especificar arquivos com base no nome de arquivo ou na extensão que deseja que o serviço exclua após um número de dias.	20 de dezembro de 2019
Políticas de ciclo de vida de objetos	Agora é possível adicionar uma regra à política de ciclo de vida do objeto que indica uma expiração por idade em segundos.	13 de setembro de 2019

[AWS CloudFormation apoio](#)

Agora você pode usar um AWS CloudFormation modelo para criar um contêiner automaticamente. O modelo do AWS CloudFormation gerencia os dados para cinco ações de API: criação de um contêiner, definição do registro de acesso, atualização da política de contêiner padrão, adição de uma política de CORS (compartilhamento de recursos entre origens) e adição de uma política de ciclo de vida de objetos.

17 de maio de 2019

[Cotas para disponibilidade de upload de streaming](#)

Para objetos com disponibilidade de upload de streaming (transferência de objetos em blocos), a operação `PutObject` não pode exceder 10 TPS e a operação `GetObject` não pode exceder 25 TPS.

8 de abril de 2019

[Transferência de objetos em blocos](#)

Adicionado suporte para transferência de objetos em blocos. Esse recurso permite especificar que um objeto está disponível para download antes de o objeto ser carregado completamente.

5 de abril de 2019

Registro de acesso	O AWS Elemental MediaStore e agora oferece suporte ao registro de acesso, que fornece registros detalhados das solicitações feitas aos objetos em um contêiner.	25 de fevereiro de 2019
Políticas de ciclo de vida de objetos	Adicionado suporte para políticas de ciclo de vida de objeto, que controlam a data de expiração de objetos dentro do contêiner atual.	12 de dezembro de 2018
Aumentada a cota de tamanho do objeto	A cota para o tamanho de um objeto agora é de 25 MB.	10 de outubro de 2018
Aumentada a cota de tamanho do objeto	A cota para o tamanho de um objeto agora é de 20 MB.	6 de setembro de 2018
AWS CloudTrail integração	O conteúdo da CloudTrail integração foi atualizado para se alinhar às mudanças recentes no CloudTrail serviço.	12 de julho de 2018
Colaboração da CDN	Foram adicionadas informações sobre como usar o AWS Elemental MediaStore com uma rede de entrega de conteúdo (CDN), como a Amazon. CloudFront	14 de abril de 2018

[Configurações do CORS](#)

O AWS Elemental MediaStore e agora oferece suporte ao compartilhamento de recursos de origem cruzada (CORS), o que permite que aplicativos web de clientes carregados em um domínio interajam com recursos em um domínio diferente.

7 de fevereiro de 2018

[Novo guia e serviço](#)

Esta é a versão inicial do serviço de originação e armazenamento de vídeo, AWS MediaStore Elemental, e do Guia do Usuário do AWS MediaStore Elemental.

27 de novembro de 2017

Note

- Os Serviços de AWS Mídia não foram projetados ou destinados ao uso com aplicativos ou em situações que exijam desempenho à prova de falhas, como operações de segurança de vida, sistemas de navegação ou comunicação, controle de tráfego aéreo ou máquinas de suporte de vida nas quais a indisponibilidade, interrupção ou falha dos serviços possa levar à morte, ferimentos pessoais, danos materiais ou danos ambientais.

AWS Glossário

Para obter a AWS terminologia mais recente, consulte o [AWS glossário](#) na Glossário da AWS Referência.