



Panduan Pengguna

# AWS Elemental MediaStore



# AWS Elemental MediaStore: Panduan Pengguna

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan kekayaan masing-masing pemiliknya, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau mungkin tidak.

---

# Table of Contents

Apa itu MediaStore? .....	1
Konsep dan terminologi .....	1
Layanan terkait .....	3
Mengakses MediaStore .....	3
Harga .....	4
Wilayah dan titik akhir .....	4
Menyiapkan AWS Elemental MediaStore .....	6
Daftar Akun AWS .....	6
Membuat pengguna administratif .....	7
Mulai .....	8
Langkah 1: Akses AWS Elemental MediaStore .....	8
Langkah 2: Buat kontainer .....	8
Langkah 3: Unggah sebuah objek .....	9
Langkah 4: Mengakses objek .....	9
Kontainer .....	11
Aturan untuk nama kontainer .....	11
Membuat kontainer .....	11
Melihat detail kontainer .....	13
Melihat daftar kontainer .....	14
Menghapus kontainer .....	15
Kebijakan .....	16
Kebijakan kontainer .....	16
Melihat kebijakan kontainer .....	17
Mengedit kebijakan kontainer .....	18
Contoh kebijakan kontainer .....	19
Kebijakan CORS .....	26
Skenario kasus penggunaan .....	26
Menambahkan kebijakan CORS .....	27
Melihat kebijakan CORS .....	28
Mengedit kebijakan CORS .....	29
Menghapus kebijakan CORS .....	30
Pemecahan Masalah .....	31
Contoh kebijakan CORS .....	32
Kebijakan siklus aktif objek .....	33

Komponen kebijakan siklus hidup objek .....	34
Menambahkan kebijakan siklus hidup objek .....	40
Melihat kebijakan siklus hidup objek .....	42
Mengedit kebijakan siklus hidup objek .....	43
Menghapus kebijakan siklus hidup objek .....	44
Kebijakan siklus hidup objek .....	45
Kebijakan metrik .....	49
Menambahkan kebijakan metrik .....	50
Melihat kebijakan metrik .....	50
Mengedit Kebijakan .....	51
Contoh kebijakan metrik .....	51
Folder .....	55
Aturan nama folder .....	55
Membuat folder .....	56
Menghapus folder .....	56
Objek .....	57
Meng-unggah Objek .....	57
Melihat daftar .....	59
Melihat detail objek .....	61
Mengunduh objek .....	62
Menghapus objek .....	64
Menghapus satu objek .....	64
Mengosongkan wadah .....	65
Keamanan .....	66
Perlindungan data .....	67
Enkripsi data .....	68
Pengelolaan Identitas dan Akses .....	68
Audiens .....	69
Mengautentikasi dengan identitas .....	69
Mengelola kebijakan menggunakan akses .....	73
Bagaimana AWS Elemental MediaStore bekerja dengan IAM .....	76
Contoh kebijakan berbasis identitas .....	84
Pemecahan Masalah .....	87
Pencatatan dan pemantauan .....	89
CloudWatch Alarm Amazon .....	89
Log AWS CloudTrail .....	90

AWS Trusted Advisor .....	90
Validasi kepatuhan .....	90
Ketahanan .....	91
Keamanan Infrastruktur .....	92
Pencegahan confused deputy lintas layanan .....	92
Pemantauan dan pencatatan .....	95
Pencatatan log panggilan API log dengan CloudTrail .....	96
MediaStoreInformasi di CloudTrail .....	96
Contoh: Entri file log .....	98
Pemantauan CloudWatch dengan .....	99
CloudWatch Log .....	100
CloudWatch Event .....	110
Metrik CloudWatch .....	113
Penandaan .....	118
Sumber daya yang didukung di AWS Elemental MediaStore .....	119
Konvensi penamaan dan penggunaan tag .....	119
Mengelola tag .....	119
Bekerja dengan CDN .....	121
Memungkinkan CloudFront untuk mengakses kontainer Anda .....	121
Menggunakan Origin Access Control (OAC) .....	122
Menggunakan rahasia bersama .....	122
MediaStoreinteraksi dengan cache HTTP .....	125
Permintaan bersyarat .....	125
Quotas .....	127
Informasi terkait .....	130
Riwayat dokumen .....	131
AWSGlosarium .....	136
.....	cxxxvii

# Apa itu AWS Elemental? MediaStore

AWS Elemental MediaStore adalah layanan originasi dan penyimpanan video yang menawarkan kinerja tinggi dan konsistensi langsung yang diperlukan untuk originasi langsung. Dengan MediaStore, Anda dapat mengelola aset video sebagai objek dalam wadah untuk membangun alur kerja media berbasis cloud yang dapat diandalkan.

Untuk menggunakan layanan, Anda mengunggah objek dari sumber, seperti encoder atau umpan data, ke wadah yang Anda buat. MediaStore

MediaStore adalah pilihan tepat untuk menyimpan file video yang terfragmentasi saat Anda membutuhkan konsistensi yang kuat, membaca dan menulis latensi rendah, dan kemampuan untuk menangani permintaan bersamaan dengan volume tinggi. Jika Anda tidak mengirimkan video streaming langsung, pertimbangkan untuk menggunakan [Amazon Simple Storage Service \(Amazon S3\)](#).

## Topik

- [MediaStore Konsep dan terminologi AWS Elemental](#)
- [Layanan terkait](#)
- [Mengakses AWS Elemental MediaStore](#)
- [Harga untuk AWS Elemental MediaStore](#)
- [Wilayah dan titik akhir untuk AWS Elemental MediaStore](#)

## MediaStore Konsep dan terminologi AWS Elemental

### ARN

[Nama Sumber Daya Amazon](#).

### Tubuh

Data yang akan diunggah ke objek.

### Rentang (Byte)

Sebuah subset dari data objek yang akan ditangani. Untuk informasi selengkapnya, lihat [rentang](#) dari spesifikasi HTTP.

## Kontainer

Namespace yang menyimpan objek. Container memiliki titik akhir yang dapat Anda gunakan untuk menulis dan mengambil objek dan melampirkan kebijakan akses.

## Titik Akhir

Titik masuk ke MediaStore layanan, diberikan sebagai URL root HTTPS.

## ETag

[Tag entitas](#), yang merupakan hash dari data objek.

## Folder

Pembagian wadah. Folder dapat menyimpan objek dan folder lainnya.

## Item

Istilah yang digunakan untuk merujuk ke objek dan folder.

## Objek

Aset, mirip dengan objek [Amazon S3](#). Objek adalah entitas fundamental yang disimpan di dalamnya MediaStore. Layanan menerima semua jenis file.

## Layanan Originasi

MediaStore dianggap sebagai layanan originasi karena merupakan titik distribusi untuk pengiriman konten media.

## Jalur

Pengidentifikasi unik untuk objek atau folder, yang menunjukkan lokasinya di wadah.

## Bagian

Sebuah subset dari data (chunk) dari suatu objek.

## Kebijakan

[Kebijakan IAM](#).

## Resource

Entitas di AWS yang dapat Anda gunakan. Setiap sumber daya AWS diberi Nama Sumber Daya Amazon (ARN) yang bertindak sebagai pengidentifikasi unik. Di MediaStore, ini adalah sumber daya dan format ARN-nya:

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

## Layanan terkait

- Amazon CloudFront adalah layanan jaringan pengiriman konten (CDN) global yang mengirimkan data dan video secara aman kepada pemirsa Anda. Gunakan CloudFront untuk menyampaikan konten dengan kinerja terbaik. Untuk informasi selengkapnya, lihat [Panduan CloudFront Pengembang Amazon](#).
- AWS CloudFormation adalah layanan yang membantu Anda memodelkan dan mengatur AWS sumber daya Anda. Anda membuat template yang menjelaskan semua AWS sumber daya yang Anda inginkan (seperti MediaStore kontainer), dan AWS CloudFormation mengurus penyediaan dan konfigurasi sumber daya tersebut untuk Anda. Anda tidak perlu membuat dan mengonfigurasi AWS sumber daya secara individual dan mencari tahu apa yang bergantung pada apa; AWS CloudFormation menangani semua itu. Untuk informasi selengkapnya, lihat [Panduan Pengguna AWS CloudFormation](#).
- AWS CloudTrail adalah layanan yang memungkinkan Anda memantau panggilan yang dilakukan ke CloudTrail API untuk akun Anda, termasuk panggilan yang dilakukan oleh AWS Management Console, AWS CLI, dan layanan lainnya. Untuk informasi selengkapnya, lihat [Panduan Pengguna AWS CloudTrail](#).
- Amazon CloudWatch adalah layanan pemantauan untuk sumber daya AWS Cloud dan aplikasi yang Anda jalankan AWS. Gunakan CloudWatch Peristiwa untuk melacak perubahan status kontainer dan objek di MediaStore. Untuk informasi selengkapnya, lihat [CloudWatch dokumentasi Amazon](#).
- AWS Identity and Access Management (IAM) adalah layanan web yang membantu Anda mengontrol akses ke sumber daya AWS secara aman bagi pengguna Anda. Gunakan IAM untuk mengontrol siapa yang dapat menggunakan AWS sumber daya Anda (otentikasi) dan sumber daya apa yang dapat digunakan pengguna dengan cara apa (otorisasi). Untuk informasi selengkapnya, lihat [Menyiapkan AWS Elemental MediaStore](#).
- Amazon Simple Storage Service (Amazon S3) adalah penyimpanan objek yang dibuat untuk menyimpan dan mengambil sejumlah data dari mana saja. Untuk informasi lebih lanjut, lihat [Dokumentasi Amazon S3](#).

## Mengakses AWS Elemental MediaStore

Anda dapat mengakses MediaStore menggunakan salah satu metode berikut:



- AWS Management Console - Prosedur di seluruh panduan ini menjelaskan cara menggunakan AWS Management Console untuk melakukan tugas MediaStore. Untuk mengakses MediaStore menggunakan konsol:

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

- AWS Command Line Interface— Untuk informasi selengkapnya, lihat [Panduan AWS Command Line Interface Pengguna](#). Untuk mengakses MediaStore menggunakan titik akhir CLI:

```
aws mediastore
```

- MediaStore API — Jika Anda menggunakan bahasa pemrograman yang tidak tersedia untuk SDK, lihat [Referensi AWS Elemental MediaStore API](#) untuk informasi tentang tindakan API dan tentang cara membuat permintaan API. Untuk mengakses MediaStore menggunakan endpoint REST API:

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

- AWS SDK — Jika Anda menggunakan bahasa pemrograman yang AWS menyediakan SDK, Anda dapat menggunakan SDK untuk mengakses MediaStore SDK menyederhanakan otentikasi, terintegrasi dengan mudah dengan lingkungan pengembangan Anda, dan menyediakan akses mudah ke perintah. MediaStore Untuk informasi lebih lanjut, lihat [Alat untuk Amazon Web Services](#).
- AWS Tools untuk Windows PowerShell — Untuk informasi selengkapnya, lihat [Panduan AWS Tools for Windows PowerShell Pengguna](#).

## Harga untuk AWS Elemental MediaStore

Seperti AWS produk lainnya, tidak ada kontrak atau komitmen minimum untuk digunakan MediaStore. Anda dikenakan biaya konsumsi per GB ketika konten masuk ke layanan dan biaya per GB bulanan untuk konten yang Anda simpan di layanan. Untuk informasi selengkapnya, lihat [Harga AWS Elemental MediaStore](#).

## Wilayah dan titik akhir untuk AWS Elemental MediaStore

Untuk mengurangi latensi data dalam aplikasi Anda, MediaStore tawarkan titik akhir regional untuk membuat permintaan Anda:

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

Untuk melihat daftar lengkap Wilayah AWS jika MediaStore tersedia, lihat [MediaStore titik akhir dan kuota AWS Elemental di Referensi Umum AWS](#).

# Menyiapkan AWS Elemental MediaStore

Bagian ini memandu Anda melalui langkah-langkah yang diperlukan untuk mengonfigurasi pengguna agar mengakses AWS Elemental MediaStore. Untuk latar belakang dan informasi tambahan tentang identitas dan manajemen akses MediaStore, lihat [Identity and Access Management untuk AWS Elemental MediaStore](#).

Untuk mulai menggunakan AWS Elemental MediaStore, selesaikan langkah-langkah berikut.

Topik

- [Daftar Akun AWS](#)
- [Membuat pengguna administratif](#)

## Daftar Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar Akun AWS

1. Buka <https://portal.aws.amazon.com/billing/signup>.
2. Ikuti petunjuk secara online.

Anda akan diminta untuk menerima panggilan telepon dan memasukkan kode verifikasi pada keypad telepon sebagai bagian dari prosedur pendaftaran.

Saat Anda mendaftar Akun AWS, Pengguna root akun AWS dibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya dalam akun. Sebagai praktik terbaik keamanan, [tetapkan akses administratif ke pengguna administratif](#), dan hanya gunakan pengguna root untuk melakukan [tugas yang memerlukan akses pengguna root](#).

AWS akan mengirimkan email konfirmasi kepada Anda setelah proses pendaftaran selesai. Anda dapat melihat aktivitas akun saat ini dan mengelola akun dengan mengunjungi <https://aws.amazon.com/> dan memilih Akun Saya.

# Membuat pengguna administratif

Setelah Anda mendaftarkan Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

## Mengamankan Pengguna root akun AWS Anda

1. Masuk ke [AWS Management Console](#) sebagai pemilik akun dengan memilih Pengguna root dan memasukkan alamat email Akun AWS Anda. Pada halaman berikutnya, masukkan kata sandi Anda.

Untuk bantuan masuk menggunakan pengguna root, lihat [Masuk sebagai pengguna root](#) dalam Panduan Pengguna AWS Sign-In.

2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

Untuk petunjuk, lihat [Mengaktifkan perangkat MFA virtual untuk pengguna root Akun AWS Anda \(konsol\)](#) dalam Panduan Pengguna IAM.

## Membuat pengguna administratif

1. Aktifkan Pusat Identitas IAM.

Untuk petunjuk, lihat [Mengaktifkan AWS IAM Identity Center](#) di Panduan AWS IAM Identity Center Pengguna.

2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna administratif.

Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat [Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM](#) di Panduan AWS IAM Identity Center Pengguna.

## Masuk sebagai pengguna administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email Anda saat Anda membuat pengguna Pusat Identitas IAM.

Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat [Masuk ke portal akses AWS](#) dalam Panduan Pengguna AWS Sign-In.

# Memulai AWS Elemental MediaStore

Tutorial Memulai ini menunjukkan cara menggunakan AWS Elemental MediaStore untuk membuat kontainer dan mengunggah objek.

Topik

- [Langkah 1: Akses AWS Elemental MediaStore](#)
- [Langkah 2: Buat kontainer](#)
- [Langkah 3: Unggah sebuah objek](#)
- [Langkah 4: Mengakses objek](#)

## Langkah 1: Akses AWS Elemental MediaStore

Setelah menyiapkan akun AWS dan membuat pengguna dan peran, Anda masuk ke konsol untuk AWS Elemental MediaStore.

Untuk mengakses AWS Elemental MediaStore

- Masuk ke AWS Management Console dan buka MediaStore konsol di <https://console.aws.amazon.com/mediastore/>.

### Note

Anda dapat login menggunakan salah satu kredensi IAM yang telah Anda buat untuk akun ini. Untuk informasi tentang membuat kredensi IAM, lihat [Menyiapkan AWS Elemental MediaStore](#).

## Langkah 2: Buat kontainer

Anda menggunakan kontainer di AWS Elemental MediaStore untuk menyimpan folder dan objek Anda. Anda dapat menggunakan kontainer untuk mengelompokkan objek terkait dengan cara yang sama dengan Anda menggunakan direktori untuk mengelompokkan file dalam sistem file. Anda tidak dikenai biaya saat membuat kontainer; Anda hanya dikenai biaya saat mengunggah objek ke kontainer.

## Untuk membuat kontainer

1. Pada halaman Container, pilih Create container.
2. Untuk Nama kontainer, ketikkan nama untuk kontainer Anda. Untuk informasi selengkapnya, lihat [Aturan untuk nama kontainer](#).
3. Pilih Buat wadah. AWS Elemental MediaStore menambahkan kontainer baru ke daftar kontainer. Awalnya, status wadah adalah Creating, dan kemudian berubah menjadi Active.

## Langkah 3: Unggah sebuah objek

Anda dapat mengunggah objek (masing-masing hingga 25 MB) ke wadah atau ke folder di dalam wadah. Untuk mengunggah objek ke folder, Anda menentukan jalur ke folder. Jika folder sudah ada, AWS Elemental MediaStore menyimpan objek dalam folder. Jika folder tidak ada, layanan membuatnya, dan kemudian menyimpan objek dalam folder.

### Note

Nama file hanya dapat berisi huruf, angka, titik (.) garis bawah (.) garis bawah, tildes (~), dan tanda hubung (-).

## Untuk mengunggah objek

1. Pada halaman kontainer, pilih nama kontainer yang baru saja Anda buat. Halaman detail untuk kontainer tersebut muncul.
2. Pilih Upload objek.
3. Untuk jalur Target, ketik jalur untuk folder. Sebagai contoh, premium/canada. Jika salah satu folder di jalur belum ada, AWS Elemental MediaStore membuatnya secara otomatis.
4. Untuk Object, pilih Browse.
5. Arahkan ke folder yang sesuai, dan pilih satu objek untuk diunggah.
6. Pilih Buka, lalu pilih Unggah.

## Langkah 4: Mengakses objek

Anda dapat mengunduh objek Anda ke titik akhir yang ditentukan.

1. Pada halaman kontainer, pilih nama kontainer yang memiliki objek yang ingin Anda unduh.
2. Jika objek yang ingin Anda unduh ada di subfolder, lanjutkan memilih nama folder hingga Anda melihat objeknya.
3. Pilih nama objek.
4. Pada halaman detail untuk objek, pilih Unduh.

# Kontainer di AWS ElementalMediaStore

Anda menggunakan kontainer diMediaStoreuntuk menyimpan folder dan objek Anda. Objek terkait dapat dikelompokkan dalam kontainer dengan cara yang sama dengan Anda menggunakan direktori untuk mengelompokkan file dalam sistem file. Anda tidak dikenakan biaya saat membuat kontainer; Anda hanya dikenakan biaya saat mengunggah objek ke kontainer. Untuk informasi lebih lanjut tentang biaya, lihat[AWS ElementalMediaStoreHarga](#).

## Topik

- [Aturan untuk nama kontainer](#)
- [Membuat kontainer](#)
- [Melihat detail untuk kontainer](#)
- [Melihat daftar kontainer](#)
- [Menghapus kontainer](#)

## Aturan untuk nama kontainer

Saat Anda memilih nama untuk kontainer Anda, ingat hal berikut:

- Nama harus unik dalam akun saat AWS.
- Nama dapat berisi huruf besar, huruf kecil, dan garis bawah (\_).
- Nama harus dari 1 sampai 255 karakter.
- Nilai peka huruf besar/kecil. Misalnya, Anda dapat memiliki kontainer bernamaMyContainerdan folder bernamamycontainerkarena nama-nama itu unik.
- Sebuah wadah tidak dapat diganti namanya setelah dibuat.

## Membuat kontainer

Anda dapat membuat hingga 100 kontainer untuk setiap akun AWS. Anda dapat membuat folder sebanyak yang Anda inginkan, selama mereka tidak bersarang lebih dari 10 tingkat dalam wadah. Selain itu, Anda dapat mengunggah objek sebanyak yang Anda inginkan ke setiap kontainer.



**Tip**

Anda juga dapat membuat kontainer secara otomatis dengan menggunakan AWS CloudFormation templat. Parameter AWS CloudFormation templat mengelola data untuk lima tindakan API: membuat kontainer, mengatur pencatatan akses, memperbarui kebijakan kontainer default, menambahkan kebijakan berbagi sumber daya lintas-asal (CORS), dan menambahkan kebijakan siklus hidup objek. Untuk informasi selengkapnya, lihat [Panduan Pengguna AWS CloudFormation](#).

Untuk membuat kontainer (konsol)

1. Buka MediaStore konsol di <https://console.aws.amazon.com/mediastore/>.
2. Pada Kontainer halaman, pilih **Membuat kontainer**.
3. Untuk **Kontainer nama**, masukkan nama untuk kontainer. Untuk informasi selengkapnya, lihat [Aturan untuk nama kontainer](#).
4. Pilih **Membuat kontainer**. AWS Elemental MediaStore menambahkan wadah baru ke daftar kontainer. Awalnya, status kontainer **Membuat**, dan kemudian berubah menjadi **Aktif**.

Untuk membuat wadah (AWS CLI)

- Di AWS CLI, menggunakan `create-container` perintah:

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

Contoh berikut menunjukkan nilai kembali:

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

## Melihat detail untuk kontainer

Rincian untuk kontainer termasuk kebijakan kontainer, endpoint, ARN, dan waktu pembuatan.

Untuk melihat detail kontainer (konsol)

1. Buka MediaStore konsol di <https://console.aws.amazon.com/mediastore/>.
2. Pada Kontainer halaman, pilih nama kontainer.

Halaman detail kontainer akan muncul. Halaman ini dibagi menjadi dua bagian:

- Parameter Objek bagian, yang berisi daftar objek dan folder dalam wadah.
- Parameter Kontainer bagian kebijakan, yang menunjukkan kebijakan berbasis sumber daya yang terkait dengan wadah ini. Untuk informasi tentang kebijakan sumber daya, lihat [Kebijakan kontainer](#).

Untuk melihat detail untuk kontainer (AWS CLI)

- Di AWS CLI, menggunakan `describe-container` perintah:

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

Contoh berikut menunjukkan nilai kembali:

```
{
  "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"
  }
}
```

## Melihat daftar kontainer

Anda dapat melihat daftar semua kontainer yang terkait dengan akun Anda.

Untuk menampilkan daftar kontainer (konsol)

- Buka MediaStore konsol di <https://console.aws.amazon.com/mediastore/>.

Parameter Kontainer halaman muncul, daftar semua kontainer yang terkait dengan akun Anda.

Untuk melihat daftar kontainer (AWS CLI)

- Di AWS CLI, menggunakan `list-containers` perintah.

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

Contoh berikut menunjukkan nilai kembali:

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

```
}
```

## Menghapus kontainer

Anda dapat menghapus kontainer hanya jika tidak memiliki objek.

Untuk menghapus kontainer (konsol)

1. Buka MediaStore konsol di <https://console.aws.amazon.com/mediastore/>.
2. Pada Kontainer halaman, pilih opsi di sebelah kiri nama kontainer.
3. Pilih Delete (Hapus).

Untuk menghapus kontainer (AWS CLI)

- Di AWS CLI, menggunakan `delete-container` perintah:

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

Perintah ini tidak memiliki nilai kembali.

# Kebijakan di AWS ElementalMediaStore

Anda dapat menerapkan satu atau lebih kebijakan ini ke AWS ElementalMediaStorekontainer:

- [Kebijakan kontainer](#)- Menetapkan hak akses ke semua folder dan objek dalam wadah. MediaStoremenetapkan kebijakan default yang memungkinkan pengguna untuk melakukan semuaMediaStoreoperasi pada wadah. Kebijakan ini menentukan bahwa semua operasi harus dilakukan melalui HTTPS. Setelah membuat kontainer, Anda dapat mengedit kebijakan kontainer.
- [Cross-origin resource sharing \(CORS\)](#)- Memungkinkan aplikasi web klien dari satu domain untuk berinteraksi dengan sumber daya di domain yang berbeda. MediaStoretidak menetapkan kebijakan CORS default.
- [Kebijakan metrik](#)- MemungkinkanMediaStoremengirim metrik ke AmazonCloudWatch. MediaStoretidak menetapkan kebijakan metrik default.
- [Kebijakan siklus aktif objek](#)- Mengontrol berapa lama objek tetap dalamMediaStorekontainer. MediaStoretidak menetapkan kebijakan siklus hidup objek default.

## Kebijakan kontainer di AWS ElementalMediaStore

Setiap kontainer memiliki kebijakan berbasis sumber daya yang mengatur hak akses ke semua folder dan objek dalam wadah tersebut. Kebijakan default, yang secara otomatis dilampirkan ke semua kontainer baru, memungkinkan akses ke semua AWS ElementalMediaStoreoperasi pada wadah. Ini menentukan bahwa akses ini memiliki kondisi yang membutuhkan HTTPS untuk operasi. Setelah membuat kontainer, Anda dapat mengedit kebijakan yang dilampirkan ke wadah tersebut.

Anda juga dapat menentukan[kebijakan siklus hidup objek](#)yang mengatur tanggal kedaluwarsa objek dalam sebuah wadah. Setelah objek mencapai usia maksimum yang Anda tentukan, layanan akan menghapus objek dari wadah.

Topik

- [Melihat kebijakan kontainer](#)
- [Mengedit kebijakan kontainer](#)
- [Contoh kebijakan kontainer](#)

## Melihat kebijakan kontainer

Anda dapat menggunakan konsol atau AWS CLI untuk melihat kebijakan berbasis sumber daya dari sebuah wadah.

Untuk melihat kebijakan kontainer (konsol)

1. Buka MediaStore konsol di <https://console.aws.amazon.com/mediastore/>.
2. Pada Kontainer halaman, pilih nama kontainer.

Halaman detail kontainer akan muncul. Kebijakan ditampilkan dalam Kebijakan kontainer bagian.

Untuk melihat kebijakan kontainer (AWS CLI)

- Di AWS CLI, gunakan `get-container-policy` perintah:

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

Contoh berikut menunjukkan nilai kembali:

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

```
    }
  }
]
}
}
```

## Mengedit kebijakan kontainer

Anda dapat mengedit izin dalam kebijakan kontainer default, atau Anda dapat membuat kebijakan baru yang menggantikan kebijakan default. Ini membutuhkan waktu hingga lima menit hingga kebijakan baru diterapkan.

Untuk mengedit kebijakan kontainer (konsol)

1. Buka MediaStore konsol di <https://console.aws.amazon.com/mediastore/>.
2. Pada Kontainer halaman, pilih nama kontainer.
3. Pilih Sunting kebijakan. Untuk contoh yang menunjukkan cara mengatur izin berbeda, lihat [the section called "Contoh kebijakan kontainer"](#).
4. Buat perubahan yang sesuai, lalu pilih Simpan.

Untuk mengedit kebijakan kontainer (AWS CLI)

1. Buat file yang mendefinisikan kebijakan kontainer:

```
{
  "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. DiAWS CLI, gunakan `put-container-policy` perintah:

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

Perintah ini tidak memiliki nilai kembali.

## Contoh kebijakan kontainer

Contoh berikut menunjukkan kebijakan kontainer yang dibangun untuk kelompok pengguna yang berbeda.

### Topik

- [Contoh kebijakan kontainer: Default](#)
- [Contoh kebijakan kontainer: Akses baca publik melalui HTTPS](#)
- [Contoh kebijakan kontainer: Akses baca publik melalui HTTP atau HTTPS](#)
- [Contoh kebijakan kontainer: Akses baca lintas-akun—HTTP diaktifkan](#)
- [Contoh kebijakan kontainer: Akses baca lintas akun melalui HTTPS](#)
- [Contoh kebijakan kontainer: Akses baca lintas-akun ke peran](#)
- [Contoh kebijakan kontainer: Akses penuh lintas akun ke peran](#)
- [Contoh kebijakan kontainer: Akses dibatasi ke alamat IP tertentu](#)

### Contoh kebijakan kontainer: Default

Saat Anda membuat kontainer, AWS ElementalMediaStore secara otomatis melampirkan kebijakan berbasis sumber daya berikut:

```
{  
  "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" }
    }
  }
]
}

```

Kebijakan ini dibangun ke dalam layanan, sehingga Anda tidak perlu membuatnya. Namun, Anda bisa [edit kebijakan](#) pada kontainer jika izin dalam kebijakan default tidak sejajar dengan izin yang ingin Anda gunakan untuk kontainer.

Kebijakan default yang ditetapkan ke semua kontainer baru memungkinkan akses ke semua MediaStore operasi pada wadah. Ini menentukan bahwa akses ini memiliki kondisi yang membutuhkan HTTPS untuk operasi.

### Contoh kebijakan kontainer: Akses baca publik melalui HTTPS

Kebijakan contoh ini memungkinkan pengguna untuk mengambil objek melalui permintaan HTTPS. Ini memungkinkan akses baca ke siapa saja melalui koneksi SSL/TLS yang aman: pengguna yang diautentikasi dan pengguna anonim (pengguna yang tidak masuk). Pernyataan itu memiliki nama `PublicReadOverHttps`. Hal ini memungkinkan akses ke `GetObject` dan `DescribeObject` operasi pada objek apapun (seperti yang ditentukan oleh \* di akhir jalur sumber daya). Ini menentukan bahwa akses ini memiliki kondisi yang membutuhkan HTTPS untuk operasi:

```

{
  "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"
    }
  }
}
]
}

```

## Contoh kebijakan kontainer: Akses baca publik melalui HTTP atau HTTPS

Contoh kebijakan ini mengizinkan akses ke `GetObject` dan `DescribeObject` operasi pada objek apapun (seperti yang ditentukan oleh \* di akhir jalur sumber daya). Ini memungkinkan akses baca ke siapa saja, termasuk semua pengguna yang diautentikasi dan pengguna anonim (pengguna yang tidak masuk):

```

{
  "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"] }
      }
    }
  ]
}

```

## Contoh kebijakan kontainer: Akses baca lintas-akun—HTTP diaktifkan

Contoh kebijakan ini memungkinkan pengguna untuk mengambil objek melalui permintaan HTTP. Ini memungkinkan akses ke pengguna yang diautentikasi dengan akses lintas akun. Objek tidak diperlukan untuk di-host pada server dengan sertifikat 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" ]
      }
    }
  } ]
}

```

## Contoh kebijakan kontainer: Akses baca lintas akun melalui HTTPS

Contoh kebijakan ini mengizinkan akses ke `GetObject` dan `DescribeObject` operasi pada objek apapun (sebagaimana ditentukan oleh \* di akhir jalur sumber daya) yang dimiliki oleh pengguna pengguna root dari yang ditentukan <other acct number>. Ini menentukan bahwa akses ini memiliki kondisi yang membutuhkan HTTPS untuk operasi:

```

{
  "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"
        }
      }
    }
  ]
}

```

## Contoh kebijakan kontainer: Akses baca lintas-akun ke peran

Contoh kebijakan mengizinkan akses ke `GetObject` dan `DescribeObject` operasi pada objek apapun (sebagaimana ditentukan oleh \* pada akhir jalur sumber daya) yang dimiliki oleh <owner acct number>. Hal ini memungkinkan akses ini ke setiap pengguna <other acct number> jika akun yang telah diasumsikan peran yang ditentukan dalam <role name>:

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

## Contoh kebijakan kontainer: Akses penuh lintas akun ke peran

Kebijakan contoh ini memungkinkan akses lintas-akun untuk memperbarui objek apa pun di akun, selama pengguna login melalui HTTP. Ini juga memungkinkan akses lintas akun untuk menghapus, mengunduh, dan menggambarkan objek melalui HTTP atau HTTPS ke akun yang telah mengasumsikan peran yang ditentukan:

- Pernyataan pertama adalah `CrossAccountRolePostOverHttps`. Hal ini memungkinkan akses ke `PutObject` operasi pada objek apapun dan memungkinkan akses ini ke setiap pengguna dari akun yang ditentukan jika akun yang telah diasumsikan peran yang ditentukan dalam <role name>. Ini menentukan bahwa akses ini memiliki kondisi yang membutuhkan HTTPS untuk operasi (kondisi ini harus selalu disertakan saat memberikan akses ke `PutObject`).

Dengan kata lain, setiap kepala sekolah yang memiliki akses lintas akun dapat mengakses `PutObject`, tetapi hanya lebih dari HTTPS.

- Pernyataan kedua adalah `CrossAccountFullAccessExceptPost`. Hal ini memungkinkan akses ke semua operasi kecuali `PutObject` pada objek apapun. Hal ini memungkinkan akses ini ke setiap pengguna dari akun yang ditentukan jika akun yang telah diasumsikan peran yang

ditentukan dalam `<role name>`. Akses ini tidak memiliki kondisi yang membutuhkan HTTPS untuk operasi.

Dengan kata lain, akun yang memiliki akses lintas akun dapat mengakses `DeleteObject`, `GetObject`, dan sebagainya (tapi tidak `PutObject`), dan dapat melakukan ini melalui HTTP atau HTTPS.

Jika Anda tidak mengecualikan `PutObject` dari pernyataan kedua, pernyataan tidak akan valid (karena jika Anda menyertakan `PutObject` Anda harus secara eksplisit mengatur HTTPS sebagai suatu kondisi).

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

## Contoh kebijakan kontainer: Akses dibatasi ke alamat IP tertentu

Contoh kebijakan ini mengizinkan akses ke semua AWS ElementalMediaStore operasi pada objek dalam wadah yang ditentukan. Namun, permintaan harus berasal dari rentang alamat IP yang ditentukan dalam syarat.

Syarat dalam pernyataan ini mengidentifikasi rentang 198.51.100.\* dari alamat IP Protokol Internet versi 4 (IPv4) yang diizinkan, dengan satu pengecualian: 198.51.100.188.

ParameterCondition blok menggunakan IpAddress dan NotIpAddress kondisi dan kondisi aws:SourceIp kondisi kunci, yang merupakan kunci kondisi AWS-lebar. Nilai IPv4 aws:sourceIp menggunakan notasi CIDR standar. Untuk informasi selengkapnya, lihat [Operator Kondisi Alamat IP](#) dalam Panduan Pengguna 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"
        }
      }
    }
  ]
}
```

# Kebijakan cross-origin resource sharing (CORS) di AWS ElementalMediaStore

Cross-origin resource sharing (CORS) menentukan cara aplikasi web klien yang dimuat dalam satu domain untuk berinteraksi dengan sumber daya di domain yang berbeda. Dengan dukungan CORS di AWS ElementalMediaStore, Anda dapat membangun aplikasi web sisi klien yang kaya denganMediaStore dan secara selektif mengizinkan akses lintas asal keMediaStore sumber daya.

## Note

Jika Anda menggunakan AmazonCloudFront untuk mendistribusikan konten dari kontainer yang memiliki kebijakan CORS, pastikan untuk [mengonfigurasi distribusi untuk AWS ElementalMediaStore](#) (termasuk langkah untuk mengedit perilaku cache untuk mengatur CORS).

Bagian ini memberikan ikhtisar tentang CORS. Subtopik menjelaskan bagaimana Anda dapat mengaktifkan CORS menggunakan AWS ElementalMediaStore konsol, atau pemrograman menggunakan MediaStore REST API dan AWS SDK.

## Topik

- [Skenario kasus penggunaan CORS](#)
- [Menambahkan kebijakan CORS ke kontainer](#)
- [Melihat kebijakan CORS](#)
- [Mengedit kebijakan CORS](#)
- [Menghapus kebijakan CORS](#)
- [Penyelesaian masalah isu CORS](#)
- [Contoh kebijakan CORS](#)

## Skenario kasus penggunaan CORS

Berikut adalah contoh skenario untuk menggunakan CORS:

- Skenario 1: Misalkan Anda mendistribusikan video streaming langsung di AWS ElementalMediaStore wadah bernama LiveVideo. Pengguna Anda memuat titik akhir video `http://livevideo.mediastore.ap-southeast-2.amazonaws.com` dari asal tertentu

seperti `www.example.com`. Anda ingin menggunakan JavaScript pemutar video untuk mengakses video yang berasal dari kontainer ini melalui `unauthenticatedGET` dan `PUT` permintaan. Browser biasanya akan memblokir JavaScript dari mengizinkan permintaan tersebut, tetapi Anda dapat menetapkan kebijakan CORS pada kontainer Anda untuk secara eksplisit mengaktifkan permintaan ini dari `www.example.com`.

- Skenario 2: Misalkan Anda ingin meng-host live stream yang sama seperti dalam Skenario 1 dari `MediaStore` kontainer, tetapi ingin mengizinkan permintaan dari asal apapun. Anda dapat mengonfigurasi kebijakan CORS untuk mengizinkan asal wildcard (\*), sehingga permintaan dari asal mana pun dapat mengakses video.

## Menambahkan kebijakan CORS ke kontainer

Bagian ini menjelaskan cara menambahkan konfigurasi cross-origin resource sharing (CORS) ke `MediaStore` kontainer. CORS memungkinkan aplikasi web klien yang dimuat di satu domain untuk berinteraksi dengan sumber daya di domain lain.

Untuk mengonfigurasi kontainer Anda agar permintaan lintas asal dapat dilakukan, Anda menambahkan kebijakan CORS ke kontainer. Kebijakan CORS menetapkan aturan yang mengidentifikasi asal-usul yang Anda izinkan untuk mengakses kontainer Anda, metode operasi (metode HTTP) yang didukung untuk setiap asal, dan informasi khusus operasi lainnya.

Saat Anda menambahkan kebijakan CORS ke kontainer, [kebijakan kontainer](#) (yang mengatur hak akses ke wadah) terus berlaku.

Untuk menambahkan kebijakan CORS

1. Buka `MediaStore` konsol di <https://console.aws.amazon.com/mediastore/>.
2. Pada `Kontainer` Halaman, pilih nama wadah yang ingin Anda buat kebijakan CORS.

Halaman detail akan muncul.

3. Di `Kebijakan CORS` bagian, pilih `Buat kebijakan CORS`.
4. Masukkan kebijakan dalam format JSON, lalu pilih `Simpan`.

Untuk menambahkan kebijakan CORS (AWS CLI)

1. Buat file yang mendefinisikan kebijakan CORS:

```
[
```



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

2. DiAWS CLI, gunakan `put-cors-policy` perintah.

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

Perintah ini tidak memiliki nilai kembali.

## Melihat kebijakan CORS

Cross-origin resource sharing (CORS) menentukan cara aplikasi web klien yang dimuat dalam satu domain untuk berinteraksi dengan sumber daya di domain yang berbeda.

Untuk melihat kebijakan CORS

1. Buka MediaStore konsol di <https://console.aws.amazon.com/mediastore/>.
2. Pada Kontainer Halaman, pilih nama kontainer yang ingin Anda lihat kebijakan CORS.

Halaman rincian kontainer muncul, dengan kebijakan CORS di Kebijakan CORS bagian.

Untuk melihat kebijakan CORS (AWS CLI)

- DiAWS CLI, gunakan `get-cors-policy` perintah:

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

Contoh berikut menunjukkan nilai kembali:

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

## Mengedit kebijakan CORS

Cross-origin resource sharing (CORS) menentukan cara aplikasi web klien yang dimuat dalam satu domain untuk berinteraksi dengan sumber daya di domain yang berbeda.

Untuk menyunting kebijakan CORS

1. Buka MediaStore konsol di <https://console.aws.amazon.com/mediastore/>.
2. Pada Kontainer Halaman, pilih nama kontainer yang ingin Anda edit kebijakan CORS.

Halaman detail akan muncul.

3. Di Kebijakan CORS bagian, pilih Mengedit kebijakan CORS.
4. Buat perubahan pada kebijakan, dan kemudian pilih Simpan.

Untuk mengedit kebijakan CORS (AWS CLI)

1. Buat file yang mendefinisikan kebijakan CORS yang diperbarui:

```
[
```

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

2. DiAWS CLI, gunakan `put-cors-policy` perintah.

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

Perintah ini tidak memiliki nilai kembali.

## Menghapus kebijakan CORS

Cross-origin resource sharing (CORS) menentukan cara aplikasi web klien yang dimuat dalam satu domain untuk berinteraksi dengan sumber daya di domain yang berbeda. Menghapus kebijakan CORS dari kontainer akan menghapus izin untuk permintaan lintas asal.

Untuk menghapus kebijakan CORS

1. Buka MediaStore konsol di <https://console.aws.amazon.com/mediastore/>.
2. Pada Kontainer Halaman, pilih nama kontainer yang ingin Anda hapus kebijakan CORS.

Halaman detail akan muncul.

3. Di Kebijakan CORS bagian, pilih **Menghapus kebijakan CORS**.
4. Memilih **Lanjutkan** untuk mengkonfirmasi, dan kemudian memilih **Simpan**.

Untuk menghapus kebijakan CORS (AWS CLI)

- DiAWS CLI, gunakan `delete-cors-policy` perintah:

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

Perintah ini tidak memiliki nilai kembali.

## Penyelesaian masalah isu CORS

Jika Anda mengalami perilaku tak terduga saat mengakses kontainer yang memiliki kebijakan CORS, ikuti langkah-langkah berikut untuk memecahkan masalah.

1. Verifikasi bahwa kebijakan CORS dilampirkan ke kontainer.

Untuk petunjuk, lihat [the section called “Melihat kebijakan CORS”](#).

2. Catat permintaan dan respons lengkap menggunakan alat pilihan Anda (seperti konsol pengembang browser Anda). Pastikan kebijakan CORS yang dilampirkan pada kontainer mencakup setidaknya satu aturan CORS yang sesuai dengan data dalam permintaan Anda, sebagai berikut:

- a. Verifikasi bahwa permintaan memiliki `Origin` header.

Jika header hilang, AWS Elemental MediaStore tidak memperlakukan permintaan tersebut sebagai permintaan lintas asal dan tidak mengirimkan header respons CORS dalam respons.

- b. Verifikasi bahwa `Origin` header dalam permintaan Anda cocok setidaknya salah satu `AllowedOrigin` elemen dalam spesifik `CORSRule`.

Skema, host, dan nilai port di `Origin` header permintaan harus sesuai dengan `AllowedOrigins` di dalam `CORSRule`. Misalnya, jika Anda mengatur `CORSRule` untuk memungkinkan asal `http://www.example.com`, maka keduanya `https://www.example.com` dan `http://www.example.com:80` origins dalam permintaan Anda tidak cocok dengan asal yang diizinkan dalam konfigurasi Anda.

- c. Verifikasi bahwa metode dalam permintaan Anda (atau metode yang ditentukan dalam `Access-Control-Request-Method` dalam kasus permintaan preflight) adalah salah satu `AllowedMethod` elemen yang sama `CORSRule`.

- d. Untuk permintaan preflight, jika permintaan mencakup header `Access-Control-Request-Headers`, verifikasi bahwa `CORSRule` termasuk entri `AllowedHeaders` untuk setiap nilai dalam header `Access-Control-Request-Headers`.

## Contoh kebijakan CORS

Contoh berikut menunjukkan kebijakan cross-origin resource sharing (CORS).

### Topik

- [Contoh kebijakan CORS: Akses baca untuk domain apa pun](#)
- [Contoh kebijakan CORS: Membaca akses untuk domain tertentu](#)

### Contoh kebijakan CORS: Akses baca untuk domain apa pun

Kebijakan berikut memungkinkan halaman web dari domain apa pun untuk mengambil konten dari AWS Elemental AndaMediaStorekontainer. Permintaan mencakup semua header HTTP dari domain asal, dan layanan hanya merespon permintaan HTTP GET dan HTTP HEAD dari domain asal. Hasilnya di-cache selama 3.000 detik sebelum serangkaian hasil baru dikirimkan.

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

### Contoh kebijakan CORS: Membaca akses untuk domain tertentu

Kebijakan berikut memungkinkan halaman web dari `https://www.example.com` untuk mengambil konten dari AWS ElementalMediaStorekontainer. Permintaan mencakup semua header HTTP

dari `https://www.example.com`, dan layanan hanya merespon permintaan HTTP GET dan HTTP HEAD dari `https://www.example.com`. Hasilnya di-cache selama 3.000 detik sebelum serangkaian hasil baru dikirimkan.

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

## Kebijakan siklus hidup objek di AWS ElementalMediaStore

Untuk setiap kontainer, Anda dapat membuat kebijakan siklus hidup objek yang mengatur berapa lama objek harus disimpan dalam wadah. Saat objek mencapai usia maksimum yang Anda tentukan, AWS ElementalMediaStore menghapus objek. Anda dapat menghapus objek setelah tidak lagi diperlukan untuk menghemat biaya penyimpanan.

Anda juga dapat menentukan MediaStore harus memindahkan objek ke kelas penyimpanan akses jarang (IA) setelah mereka mencapai usia tertentu. Objek yang disimpan di kelas penyimpanan IA memiliki tarif yang berbeda untuk penyimpanan dan pengambilan dari objek yang disimpan di kelas penyimpanan standar. Untuk informasi selengkapnya, lihat [MediaStore Harga](#).

Kebijakan siklus hidup objek berisi aturan, yang mendikte umur objek oleh subfolder. (Anda tidak dapat menetapkan kebijakan siklus hidup objek ke masing-masing objek). Anda hanya dapat melampirkan satu kebijakan siklus hidup objek ke kontainer, tetapi Anda dapat menambahkan hingga 10 aturan ke setiap kebijakan siklus hidup objek. Untuk informasi selengkapnya, lihat [Komponen kebijakan siklus hidup objek](#).

### Topik

- [Komponen kebijakan siklus hidup objek](#)

- [Menambahkan kebijakan siklus hidup objek ke kontainer](#)
- [Melihat kebijakan siklus hidup objek](#)
- [Mengedit kebijakan siklus hidup objek](#)
- [Menghapus kebijakan siklus hidup objek](#)
- [Kebijakan siklus hidup objek](#)

## Komponen kebijakan siklus hidup objek

Kebijakan siklus hidup objek mengatur berapa lama objek tetap berada di AWS ElementalMediaStorekontainer. Setiap kebijakan siklus hidup terdiri dari satu atau lebih aturan, yang menentukan umur objek. Aturan dapat berlaku untuk satu folder, beberapa folder, atau seluruh kontainer.

Anda dapat melampirkan satu kebijakan siklus hidup objek ke kontainer, dan setiap kebijakan siklus hidup objek dapat berisi hingga 10 aturan. Anda tidak dapat menetapkan kebijakan siklus hidup objek ke objek individual.

### Aturan dalam kebijakan siklus hidup objek

Anda dapat membuat tiga jenis aturan:

- [Data transien](#)
- [Hapus objek](#)
- [Transisi siklus hidup](#)

#### Data transien

Aturan data transien menetapkan objek untuk kedaluwarsa dalam hitungan detik. Jenis aturan ini hanya berlaku untuk objek yang ditambahkan ke wadah setelah kebijakan menjadi efektif. Butuh waktu hingga 20 menit untukMediaStoreuntuk menerapkan kebijakan baru ke wadah.

Contoh aturan untuk data sementara terlihat seperti ini:

```
{
  "definition": {
    "path": [ {"wildcard": "Football/index*.m3u8"} ],
    "seconds_since_create": [
```

```
        {"numeric": [ ">", 120 ]}
      ],
      "action": "EXPIRE"
    },
```

Aturan data transien memiliki tiga bagian:

- **path:** Selalu diatur ke wildcard. Anda menggunakan bagian ini untuk menentukan objek yang ingin Anda hapus. Anda dapat menggunakan satu atau beberapa wildcard, yang diwakili oleh tanda bintang (\*). Setiap wildcard mewakili kombinasi nol karakter atau lebih. Misalnya, "path": [ {"wildcard": "Football/index\*.m3u8"} ], berlaku untuk semua file dalam Football folder yang cocok dengan pola index\*.m3u8 (seperti index.m3u8, index1.m3u8, dan index123456.m3u8). Anda dapat menyertakan hingga 10 jalur dalam satu aturan.
- **seconds\_since\_create:** Selalu diatur ke numeric. Anda dapat menentukan nilai dari 1-300 detik. Anda juga dapat mengatur operator ke lebih besar dari (>) atau lebih besar dari atau sama dengan (>=).
- **action:** Selalu diatur ke EXPIRE.

Untuk aturan data sementara (objek berakhir dalam hitungan detik), tidak ada lag antara berakhirnya objek dan penghapusan objek.

#### Note

Objek yang tunduk pada aturan data sementara tidak termasuk dalam `list-items` tanggapan. Selain itu, objek yang kedaluwarsa karena aturan data sementara tidak memancarkan `CloudWatch` acara ketika mereka kedaluwarsa.

## Hapus objek

Aturan objek hapus menetapkan objek untuk kedaluwarsa dalam beberapa hari. Jenis aturan ini berlaku untuk semua objek dalam wadah, bahkan jika mereka ditambahkan ke wadah sebelum kebijakan dibuat. Butuh waktu hingga 20 menit untuk MediaStore untuk menerapkan kebijakan baru, tetapi membutuhkan waktu hingga 24 jam agar objek dapat dihapus dari kontainer.

Contoh dari dua aturan untuk menghapus objek terlihat seperti ini:



```

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

```

Hapus aturan objek memiliki tiga bagian:

- **path:** Atur ke salah satu prefix atau wildcard. Anda tidak bisa mencampur prefix dan wildcard dalam aturan yang sama. Jika Anda ingin menggunakan keduanya, Anda harus membuat satu aturan untuk prefix dan aturan terpisah untuk wildcard, seperti yang ditunjukkan pada contoh di atas.
- **prefix-** Anda mengatur jalan untuk prefix jika Anda ingin menghapus semua objek dalam folder tertentu. Jika parameternya kosong ("path": [ { "prefix": "" } ],), target adalah semua objek yang disimpan di mana saja dalam wadah saat ini. Anda dapat menyertakan hingga 10 prefix jalur dalam satu aturan.
- **wildcard-** Anda mengatur jalan untuk wildcard jika Anda ingin menghapus objek tertentu berdasarkan nama file dan/atau jenis file. Anda dapat menggunakan satu atau beberapa wildcard, yang diwakili oleh tanda bintang (\*). Setiap wildcard mewakili kombinasi nol karakter atau lebih. Misalnya, "path": [ {"wildcard": "Football/\*.ts"} ], berlaku untuk semua file dalam Football folder yang cocok dengan pola \*.ts (seperti nama file, filename1.ts, dan filename123456.ts). Anda dapat menyertakan hingga 10 wildcard jalur dalam satu aturan.
- **days\_since\_create:** Selalu diatur ke numeric. Anda dapat menentukan nilai dari 1-36,500 hari. Anda juga dapat mengatur operator ke lebih besar dari (>) atau lebih besar dari atau sama dengan (>=).

- `action`: Selalu diatur ke `EXPIRE`.

Untuk menghapus aturan objek (objek berakhir dalam beberapa hari), mungkin ada sedikit lag antara berakhirnya objek dan penghapusan objek. Namun, perubahan dalam penagihan terjadi segera setelah objek kedaluwarsa. Misalnya, jika aturan siklus hidup menentukan `10days_since_create`, akun tidak ditagih untuk objek setelah objek berusia 10 hari, bahkan jika objek belum dihapus.

### Transisi siklus hidup

Aturan transisi siklus hidup menetapkan objek untuk dipindahkan ke kelas penyimpanan akses jarang (IA) setelah mencapai usia tertentu, diukur dalam beberapa hari. Objek yang disimpan di kelas penyimpanan IA memiliki tarif yang berbeda untuk penyimpanan dan pengambilan dari objek yang disimpan di kelas penyimpanan standar. Untuk informasi selengkapnya, lihat [MediaStore Harga](#).

Setelah sebuah objek telah pindah ke kelas penyimpanan IA, Anda tidak dapat memindahkannya kembali ke kelas penyimpanan standar.

Aturan transisi siklus hidup berlaku untuk semua objek dalam wadah, bahkan jika mereka ditambahkan ke kontainer sebelum kebijakan dibuat. Butuh waktu hingga 20 menit untuk MediaStore untuk menerapkan kebijakan baru, tetapi dapat memakan waktu hingga 24 jam agar objek dapat dihapus dari wadah.

Contoh aturan transisi siklus hidup terlihat seperti ini:

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

Aturan transisi siklus hidup memiliki tiga bagian:

- `path`: Atur ke salah satu `prefix` atau `wildcard`. Anda tidak bisa mencampur `prefix` dan `wildcard` dalam aturan yang sama. Jika Anda ingin menggunakan keduanya, Anda harus membuat satu aturan untuk `prefix` dan aturan terpisah untuk `wildcard`.

- **prefix-** Anda mengatur jalan untuk **prefix** jika Anda ingin transisi semua objek dalam folder tertentu ke kelas penyimpanan IA. Jika parameternya kosong (`"path": [ { "prefix": "" } ],`), target adalah semua objek yang disimpan di mana saja dalam wadah saat ini. Anda dapat menyertakan hingga 10 **prefix** jalur dalam satu aturan.
- **wildcard-** Anda mengatur jalan untuk **wildcard** jika Anda ingin transisi objek tertentu ke kelas penyimpanan IA berdasarkan nama file dan/atau jenis file. Anda dapat menggunakan satu atau beberapa **wildcard**, yang diwakili oleh tanda bintang (\*). Setiap **wildcard** mewakili kombinasi nol karakter atau lebih. Misalnya, `"path": [ {"wildcard": "Football/*.ts"} ],` berlaku untuk semua file dalam `Football` folder yang cocok dengan pola `*.ts` (seperti nama file, `filename1.ts`, dan `filename123456.ts`). Anda dapat menyertakan hingga 10 **wildcard** jalur dalam satu aturan.
- **days\_since\_create:** Selalu diatur ke `"numeric": [ ">=" , 30 ]`.
- **action:** Selalu diatur ke `ARCHIVE`.

## Contoh

Misalkan sebuah wadah bernama `LiveEvents` memiliki empat subfolder: `Football`, `Baseball`, `Basketball`, dan `AwardsShow`. Kebijakan siklus hidup objek yang ditetapkan ke `LiveEvents` folder mungkin terlihat seperti ini:

```
{
  "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"
}

```

```
]
}
```

Kebijakan menentukan hal-hal berikut:

- Aturan pertama menginstruksikan AWS ElementalMediaStore untuk menghapus objek yang disimpan dalamLiveEvents/Footballfolder danLiveEvents/Baseballfolder setelah mereka lebih tua dari 28 hari.
- Aturan kedua menginstruksikan layanan untuk menghapus objek yang disimpan dalamLiveEvents/AwardsShowfolder ketika mereka berusia 15 hari atau lebih.
- Aturan ketiga menginstruksikan layanan untuk menghapus objek yang disimpan di mana saja diLiveEventskontainer setelah mereka lebih tua dari 40 hari. Aturan ini berlaku untuk objek yang disimpan langsung diLiveEventskontainer, serta benda-benda yang disimpan dalam salah satu dari empat subfolder kontainer.
- Aturan keempat menginstruksikan layanan untuk menghapus objek diFootballfolder yang cocok dengan pola\*.tssetelah mereka lebih tua dari 20 hari.
- Aturan kelima menginstruksikan layanan untuk menghapus objek diFootballfolder yang cocok dengan polaindex\*.m3u8setelah mereka lebih tua dari 15 detik. MediaStoremenghapus file-file ini 16 detik setelah mereka ditempatkan dalam wadah.
- Aturan keenam menginstruksikan layanan untuk memindahkan objek diProgramfolder ke kelas penyimpanan IA setelah mereka berusia 30 hari.

Untuk contoh kebijakan siklus hidup objek lainnya, lihat[Kebijakan siklus hidup objek](#).

## Menambahkan kebijakan siklus hidup objek ke kontainer

Kebijakan siklus hidup objek memungkinkan Anda menentukan berapa lama untuk menyimpan objek Anda dalam wadah. Anda menetapkan tanggal kedaluwarsa, dan setelah tanggal kedaluwarsa AWS ElementalMediaStoremenghapus objek. Butuh waktu hingga 20 menit agar layanan menerapkan kebijakan baru ke kontainer.

Untuk informasi tentang cara membuat kebijakan siklus hidup, lihat[Komponen kebijakan siklus hidup objek](#).

**Note**

Untuk menghapus aturan objek (objek berakhir dalam beberapa hari), mungkin ada sedikit lag antara berakhirnya objek dan penghapusan objek. Namun, perubahan dalam penagihan terjadi segera setelah objek kedaluwarsa. Misalnya, jika aturan siklus hidup menentukan `10days_since_create`, akun tidak ditagih untuk objek setelah objek berusia 10 hari, bahkan jika objek belum dihapus.

Untuk menambahkan kebijakan siklus hidup objek (konsol)

1. Buka [MediaStore konsol](https://console.aws.amazon.com/mediastore/) <https://console.aws.amazon.com/mediastore/>.
2. Pada **Kontainer Halaman**, pilih nama kontainer yang ingin Anda buat kebijakan siklus hidup objek.

Halaman detail akan muncul.

3. Di **Kebijakan siklus aktif objek bagian**, pilih **Membuat kebijakan siklus aktif objek**.
4. Masukkan kebijakan dalam format JSON, lalu pilih **Simpan**.

Untuk menambahkan kebijakan siklus hidup objek (AWS CLI)

1. Membuat file yang mendefinisikan kebijakan siklus hidup objek:

```
{
  "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. DiAWS CLI, gunakan `put-lifecycle-policy` perintah:

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

Perintah ini tidak memiliki nilai kembali. Layanan melampirkan kebijakan yang ditentukan ke wadah.

## Melihat kebijakan siklus hidup objek

Kebijakan siklus hidup objek menentukan berapa lama objek harus disimpan dalam wadah.

Untuk melihat kebijakan siklus hidup objek (konsol)

1. Buka MediaStore konsol <https://console.aws.amazon.com/mediastore/>.
2. Pada Kontainer Halaman, pilih nama kontainer yang ingin Anda lihat kebijakan siklus hidup objek.

Halaman rincian kontainer muncul, dengan kebijakan siklus hidup objek di Kebijakan siklus aktif objek bagian.

Untuk melihat kebijakan siklus hidup objek (AWS CLI)

- DiAWS CLI, gunakan `get-lifecycle-policy` perintah:

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

Contoh berikut menunjukkan nilai kembali:

```
{
```

```

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

```

## Mengedit kebijakan siklus hidup objek

Anda tidak dapat mengedit kebijakan siklus hidup objek yang ada. Namun, Anda dapat mengubah kebijakan yang ada dengan mengunggah kebijakan penggantian. Butuh waktu hingga 20 menit agar layanan menerapkan kebijakan yang diperbarui ke kontainer.

Untuk menyunting kebijakan siklus hidup objek (konsol)

1. Buka MediaStore konsol <https://console.aws.amazon.com/mediastore/>.
2. Pada Kontainer Halaman, pilih nama kontainer yang ingin Anda edit kebijakan siklus hidup objek.

Halaman detail akan muncul.

3. Di Kebijakan siklus aktif objek bagian, pilih Edit kebijakan siklus aktif objek.
4. Buat kebijakan, dan pilih Simpan.

Untuk menyunting kebijakan siklus hidup objek (AWS CLI)

1. Buat file yang mendefinisikan kebijakan siklus hidup objek yang diperbarui:

```

{
  "rules": [
    {

```



```

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

```

## 2. DiAWS CLI, gunakan `put-lifecycle-policy` perintah:

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

Perintah ini tidak memiliki nilai kembali. Layanan melampirkan kebijakan yang ditentukan ke wadah, menggantikan kebijakan sebelumnya.

## Menghapus kebijakan siklus hidup objek

Saat Anda menghapus kebijakan siklus hidup objek, dibutuhkan waktu hingga 20 menit agar layanan menerapkan perubahan pada kontainer.

Untuk menghapus kebijakan siklus hidup objek (konsol)

1. Buka MediaStore konsol <https://console.aws.amazon.com/mediastore/>.
2. Pada Kontainer Halaman, pilih nama kontainer yang ingin Anda hapus kebijakan siklus hidup objek.

Halaman detail akan muncul.

3. Di Kebijakan siklus aktif objek bagian, pilih Menghapus kebijakan siklus hidup.
4. Memiilih Lanjutkan untuk mengkonfirmasi, dan kemudian memilih Simpan.

Untuk menghapus kebijakan siklus hidup objek (AWS CLI)

- DiAWS CLI, gunakan `delete-lifecycle-policy` perintah:

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

Perintah ini tidak memiliki nilai kembali.

## Kebijakan siklus hidup objek

Contoh berikut menunjukkan kebijakan siklus hidup objek.

Topik

- [Kebijakan siklus hidup objek: Kedaluwarsa dalam hitungan detik](#)
- [Kebijakan siklus hidup objek: Kedaluwarsa dalam beberapa hari](#)
- [Contoh kebijakan siklus hidup objek: Transisi ke kelas penyimpanan akses jarang](#)
- [Contoh kebijakan siklus hidup objek: Beberapa aturan](#)
- [Contoh kebijakan siklus hidup objek: Kontainer kosong](#)

### Kebijakan siklus hidup objek: Kedaluwarsa dalam hitungan detik

Kebijakan ini menentukan bahwa MediaStore menghapus objek yang cocok dengan semua kriteria berikut:

- Objek ditambahkan ke wadah setelah kebijakan menjadi efektif.
- Objek disimpan dalam `Football` folder.
- Objek memiliki ekstensi file `3u8`.
- Objek telah berada dalam wadah selama lebih dari 20 detik.

```
{
  "rules": [
    {
      "definition": {
        "path": [
```

```

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

```

## Kebijakan siklus hidup objek: Kedaluwarsa dalam beberapa hari

Kebijakan ini menentukan bahwa MediaStore menghapus objek yang cocok dengan semua kriteria berikut:

- Objek disimpan dalam Program lipat
- Objek memiliki ekstensi file ts
- Objek telah dalam wadah selama lebih dari 5 hari

```

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

```

## Contoh kebijakan siklus hidup objek: Transisi ke kelas penyimpanan akses jarang

Kebijakan ini menentukan bahwa MediaStore memindahkan objek ke kelas penyimpanan akses jarang (IA) ketika mereka berusia 30 hari. Objek yang disimpan di kelas penyimpanan IA memiliki tarif yang

berbeda untuk penyimpanan dan pengambilan dari objek yang disimpan di kelas penyimpanan standar.

Parameter `days_since_create` bidang harus diatur ke `"numeric": [ ">=" , 30 ]`.

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

## Contoh kebijakan siklus hidup objek: Beberapa aturan

Kebijakan ini menentukan bahwa MediaStore melakukan hal berikut:

- Pindahkan objek yang disimpan dalam `AwardsShow` folder ke kelas penyimpanan akses jarang (IA) setelah 30 hari
- Hapus objek yang memiliki ekstensi `3u8` dan disimpan dalam `Football` folder setelah 20 detik
- Hapus objek yang disimpan dalam `April` folder setelah 10 hari
- Hapus objek yang memiliki ekstensi `filets` dan disimpan dalam `Program` folder setelah 5 hari

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

## Contoh kebijakan siklus hidup objek: Kontainer kosong

Kebijakan siklus hidup objek berikut menentukannya MediaStore menghapus semua objek dalam wadah, termasuk folder dan subfolder, 1 hari setelah ditambahkan ke wadah. Jika kontainer memegang objek apa pun sebelum kebijakan ini diterapkan, MediaStore menghapus objek 1 hari setelah kebijakan menjadi efektif. Butuh waktu hingga 20 menit agar layanan menerapkan kebijakan baru ke kontainer.

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

## Kebijakan metrik di AWS Elemental MediaStore

Untuk setiap kontainer, Anda dapat menambahkan kebijakan metrik untuk memungkinkan AWS Elemental MediaStore mengirim metrik ke Amazon CloudWatch. Butuh waktu hingga 20 menit agar kebijakan baru diterapkan. Untuk deskripsi masing-masing MediaStore metrik, lihat [MediaStore metrik](#).

Kebijakan metrik berisi yang berikut:

- Pengaturan untuk mengaktifkan atau menonaktifkan metrik di tingkat kontainer.
- Di mana saja dari nol hingga lima aturan yang mengaktifkan metrik pada tingkat objek. Jika kebijakan berisi aturan, setiap aturan harus menyertakan kedua hal berikut:
  - Sebuah grup obyek yang mendefinisikan obyek untuk disertakan dalam grup. Definisi dapat berupa path atau nama file, tetapi tidak dapat memiliki lebih dari 900 karakter. Karakter yang

valid adalah: a-z, a-z, 0-9, \_ (garis bawah), = (sama),: (titik),. (periode), - (tanda hubung), ~ (tilde),/(garis miring), dan \* (tanda bintang). Wildcard (\*) dapat diterima.

- Sebuah nama grup obyek yang memungkinkan Anda untuk merujuk ke grup obyek. Nama tidak boleh memiliki lebih dari 30 karakter. Karakter yang valid adalah: a-z, a-z, 0-9, dan \_ (garis bawah).

Jika objek cocok dengan beberapa aturan, CloudWatch menampilkan titik data untuk setiap aturan yang cocok. Misalnya, jika sebuah objek cocok dengan dua aturan bernama `rule1` dan `rule2`, CloudWatch menampilkan dua titik data untuk aturan ini. Yang pertama memiliki `dimensiObjectGroupName=rule1` dan yang kedua memiliki `dimensiObjectGroupName=rule2`.

## Topik

- [Menambahkan kebijakan metrik](#)
- [Melihat kebijakan metrik](#)
- [Mengedit Kebijakan](#)
- [Contoh kebijakan metrik](#)

## Menambahkan kebijakan metrik

Kebijakan metrik berisi aturan yang menentukan metrik mana yang MediaStore dikirim AWS Elemental ke Amazon CloudWatch. Untuk contoh kebijakan metrik, lihat [Contoh kebijakan metrik](#).

Untuk menambahkan kebijakan metrik (konsol)

1. Buka MediaStore konsol di <https://console.aws.amazon.com/mediastore/>.
2. Di Halaman Container, pilih nama kontainer yang ingin ditambahkan kebijakan metrik.

Halaman detail kontainer akan muncul.

3. Di bagian Kebijakan metrik, pilih Buat kebijakan metrik.
4. Masukkan kebijakan dalam format JSON, lalu pilih Simpan.

## Melihat kebijakan metrik

Anda dapat menggunakan konsol atau AWS CLI untuk melihat kebijakan metrik kontainer.

Untuk melihat kebijakan metrik (konsol)

1. Buka MediaStore konsol di <https://console.aws.amazon.com/mediastore/>.
2. Pada halaman Container, pilih nama kontainer.

Halaman detail kontainer akan muncul. Kebijakan ditampilkan di bagian Kebijakan metrik.

## Mengedit Kebijakan

Kebijakan metrik berisi aturan yang menentukan metrik mana yang MediaStore dikirim AWS Elemental ke Amazon CloudWatch. Saat Anda mengedit kebijakan metrik yang ada, kebijakan baru akan muncul hingga 20 menit agar kebijakan baru akan muncul. Untuk contoh kebijakan metrik, lihat [Contoh kebijakan metrik](#).

Untuk mengedit kebijakan metrik (konsol)

1. Buka MediaStore konsol di <https://console.aws.amazon.com/mediastore/>.
2. Pada halaman Container, pilih nama kontainer.
3. Di bagian Kebijakan metrik, pilih Edit kebijakan metrik.
4. Buat perubahan yang sesuai, dan kemudian pilih Simpan.

## Contoh kebijakan metrik

Contoh berikut menunjukkan kebijakan metrik yang dibuat untuk kasus penggunaan yang berbeda.

Topik

- [Contoh kebijakan metrik: Metrik tingkat kontainer](#)
- [Contoh kebijakan metrik: Metrik tingkat jalur](#)
- [Contoh kebijakan metrik: Tingkat kontainer dan metrik tingkat jalur](#)
- [Contoh kebijakan metrik: Metrik tingkat jalur menggunakan wildcard](#)
- [Contoh kebijakan metrik: Metrik tingkat jalur dengan aturan yang tumpang tindih](#)

### Contoh kebijakan metrik: Metrik tingkat kontainer

Contoh kebijakan ini menunjukkan bahwa AWS Elemental MediaStore harus mengirim metrik ke Amazon CloudWatch pada tingkat kontainer. Misalnya, RequestCount metrik yang menghitung



jumlahPut permintaan yang dibuat ke kontainer. Cara alternatif, Anda dapat mengatur ini keDISABLED.

Karena tidak ada aturan dalam kebijakan ini, MediaStore tidak mengirim metrik di tingkat jalur. Misalnya, Anda tidak dapat melihat berapa banyakPut permintaan yang dibuat ke folder tertentu dalam wadah ini.

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

### Contoh kebijakan metrik: Metrik tingkat jalur

Kebijakan contoh ini menunjukkan bahwa AWS Elemental tidak MediaStore boleh mengirim metrik ke Amazon CloudWatch pada tingkat kontainer. Selain itu, MediaStore harus mengirim metrik untuk objek dalam dua folder tertentu:baseball/saturday danfootball/saturday. Metrik untuk MediaStore permintaan adalah sebagai berikut:

- Permintaan kebaseball/saturday folder memiliki CloudWatch dimensiObjectGroupName=baseballGroup.
- Permintaan kefootball/saturday folder memiliki dimensiObjectGroupName=footballGroup.

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

## Contoh kebijakan metrik: Tingkat kontainer dan metrik tingkat jalur

Contoh kebijakan ini menunjukkan bahwa AWS Elemental MediaStore harus mengirim metrik ke Amazon CloudWatch pada tingkat kontainer. Selain itu, MediaStore harus mengirim metrik untuk objek dalam dua folder tertentu: `baseball/saturday` dan `football/saturday`. Metrik untuk MediaStore permintaan adalah sebagai berikut:

- Permintaan ke `baseball/saturday` folder memiliki CloudWatch dimensi `ObjectGroupName=baseballGroup`.
- Permintaan ke `football/saturday` folder memiliki CloudWatch dimensi `ObjectGroupName=footballGroup`.

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

## Contoh kebijakan metrik: Metrik tingkat jalur menggunakan wildcard

Contoh kebijakan ini menunjukkan bahwa AWS Elemental MediaStore harus mengirim metrik ke Amazon CloudWatch pada tingkat kontainer. Selain itu, juga MediaStore harus mengirim metrik untuk objek berdasarkan nama file mereka. Wildcard menunjukkan bahwa objek dapat disimpan di mana saja di wadah dan mereka dapat memiliki nama file apa pun, asalkan diakhiri dengan `.m3u8` ekstensi.

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

```
]
}
```

## Contoh kebijakan metrik: Metrik tingkat jalur dengan aturan yang tumpang tindih

Contoh kebijakan ini menunjukkan bahwa AWS Elemental MediaStore harus mengirim metrik ke Amazon CloudWatch pada tingkat kontainer. Selain itu, MediaStore harus mengirim metrik untuk dua folder: `sports/football/saturday` dan `sports/football`.

Metrik untuk MediaStore permintaan ke `sports/football/saturday` folder memiliki CloudWatch dimensi `ObjectGroupName=footballGroup1`. Karena objek yang disimpan dalam `sports/football` folder cocok dengan kedua aturan, CloudWatch menampilkan dua titik data untuk objek ini: satu dengan dimensi `ObjectGroupName=footballGroup1` dan yang kedua dengan dimensi `ObjectGroupName=footballGroup2`.

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

# Folder di AWS ElementalMediaStore

Folder adalah divisi dalam sebuah kontainer. Anda menggunakan folder untuk membagi kontainer dengan cara yang sama dengan Anda membuat subfolder untuk membagi folder dalam sistem file. Anda dapat membuat hingga 10 tingkat folder (tidak termasuk wadah itu sendiri).

Folder bersifat opsional; Anda dapat memilih untuk mengunggah objek Anda langsung ke wadah, bukan folder. Namun, folder adalah cara mudah untuk mengatur objek Anda.

Untuk mengunggah objek ke folder, Anda menentukan path ke folder. Jika folder sudah ada, AWS ElementalMediaStore menyimpan objek dalam folder. Jika folder tidak ada, layanan membuatnya, dan kemudian menyimpan objek dalam folder.

Misalnya, Anda memiliki sebuah kontainer dengan namamovies, dan Anda mengunggah file bernamamlaw.ts dengan jalanpremium/canada. AWS ElementalMediaStore menyimpan objek di subfolder canada di bawah premium folder. Jika tidak ada folder, layanan akan membuat keduapremiumfolder dancanadasubfolder, dan kemudian menyimpan objek Anda dicanadasubfolder. Jika Anda hanya menentukan wadahmovies(tanpa jalan), layanan menyimpan objek langsung dalam wadah.

AWS ElementalMediaStore secara otomatis menghapus folder saat Anda menghapus objek terakhir di folder itu. Layanan ini juga menghapus folder kosong di atas folder itu. Misalnya, misalkan Anda memiliki folder bernama premium yang tidak berisi file apa pun tetapi mengandung satu subfolder bernama canada. Parametercanadasubfolder berisi satu file bernamamlaw.ts. Jika Anda menghapus filemlaw.ts, layanan menghapus keduapremiumdancanadafolder. Penghapusan otomatis ini hanya berlaku untuk folder. Layanan ini tidak menghapus kontainer kosong.

## Topik

- [Aturan nama folder](#)
- [Membuat folder](#)
- [Menghapus folder](#)

## Aturan nama folder

Saat Anda memilih nama untuk folder Anda, ingat hal berikut:

- Nama dapat berisi karakter berikut: huruf besar (A-Z), huruf kecil (a-z), angka (0-9), tanda hubung (-), tildes (~), garis bawah (\_), tanda hubung (').
- Nama harus minimal satu karakter. Nama folder kosong (seperti `folder1//folder3/`) tidak diperbolehkan.
- Nilai peka huruf besar/kecil. Misalnya, Anda dapat memiliki folder dengan nama `myFolder` dan folder bernama `myfolder` dalam wadah atau folder yang sama karena nama-nama itu unik.
- Nama harus unik hanya dalam wadah atau folder induknya. Misalnya, Anda dapat membuat folder dengan nama `myfolder` dalam dua wadah yang berbeda: `movies/myfolder` dan `sports/myfolder`.
- Nama dapat memiliki nama yang sama dengan kontainer induknya.
- Folder tidak dapat diganti namanya setelah dibuat.

## Membuat folder

Anda dapat membuat folder saat Anda mengunggah objek. Untuk mengunggah objek ke folder, Anda menentukan path ke folder. Jika folder sudah ada, AWS Elemental MediaStore menyimpan objek dalam folder. Jika folder tidak ada, layanan membuatnya, dan kemudian menyimpan objek dalam folder.

Untuk informasi selengkapnya, lihat [the section called “Meng-unggah Objek”](#).

## Menghapus folder

Anda dapat menghapus folder hanya jika folder kosong; Anda tidak dapat menghapus folder yang berisi objek.

AWS Elemental MediaStore secara otomatis menghapus folder saat Anda menghapus objek terakhir di folder itu. Layanan ini juga menghapus folder kosong di atas folder itu. Misalnya, anggaplah Anda memiliki folder dengan nama `premium` yang tidak berisi file apa pun tetapi mengandung satu subfolder bernama `canada`. Parameter `canada` subfolder berisi satu file bernama `law.ts`. Jika Anda menghapus file `law.ts`, layanan menghapus kedua `premium` dan `canada` folder. Penghapusan otomatis ini hanya berlaku untuk folder. Layanan ini tidak menghapus kontainer kosong.

Untuk informasi selengkapnya, lihat [Menghapus objek](#).

# Objek di AWS ElementalMediaStore

AWS ElementalMediaStoreaset disebutobjek. Anda dapat mengunggah objek ke wadah atau ke folder dalam wadah.

MasukMediaStore, Anda dapat mengunggah, mengunduh, dan menghapus objek:

- Unggah- Tambahkan objek ke wadah atau folder. Ini tidak sama dengan membuat objek. Anda harus membuat objek Anda secara lokal sebelum Anda dapat mengunggahnya keMediaStore.
- Unduh— Salin objek dariMediaStoreke lokasi lain. Ini tidak menghapus objek dariMediaStore.
- Hapus— Hapus objek dariMediaStoresepenuhnya. Anda dapat menghapus objek secara terpisah, atau Anda dapat[menambahkan kebijakan siklus hidup objek](#)untuk secara otomatis menghapus objek dalam wadah setelah durasi tertentu.

MediaStoremenerima semua jenis file.

Topik

- [Meng-unggah Objek](#)
- [Melihat daftar objek](#)
- [Melihat detail objek](#)
- [Mengunduh objek](#)
- [Penghapusan objek](#)

## Meng-unggah Objek

Anda dapat mengunggah objek ke wadah atau ke folder dalam wadah. Untuk mengunggah objek ke folder, Anda menentukan path ke folder. Jika folder sudah ada, AWS ElementalMediaStoremenyimpan objek dalam folder. Jika folder tidak ada, layanan membuatnya, dan kemudian menyimpan objek dalam folder. Untuk informasi lebih lanjut tentang folder, lihat[Folder di AWS ElementalMediaStore](#).

Anda dapat menggunakanMediaStorekonsol atauAWS CLIuntuk meng-unggah objek.

MediaStoremendukung transfer chunked objek, yang mengurangi latensi dengan membuat objek yang tersedia untuk diunduh saat masih diunggah. Untuk menggunakan kemampuan

ini, atur ketersediaan upload objek ke streaming. Anda dapat mengatur nilai header ini saat Anda [mengunggah objek menggunakan API](#). Jika Anda tidak menentukan header ini dalam permintaan Anda, MediaStore memberikan nilai default standar untuk ketersediaan upload objek.

Ukuran objek tidak dapat melebihi 25 MB untuk ketersediaan upload standar dan 10 MB untuk ketersediaan upload streaming.

#### Note

Nama file objek hanya dapat berisi huruf, angka, titik (.), garis bawah (\_), tildes (~), tanda hubung (-), tanda hubung (=), dan titik dua (:), dan titik dua (:).

Untuk mengunggah objek (konsol)

1. Buka MediaStore konsol di <https://console.aws.amazon.com/mediastore/>.
2. Pada Kontainer halaman, memilih nama wadah. Panel rincian untuk wadah muncul.
3. Pilih Meng-unggah objek.
4. Untuk Jalur target, ketik path untuk folder. Misalnya, premium/canada. Jika salah satu folder di jalur yang Anda tentukan belum ada, layanan akan membuatnya secara otomatis.
5. Di Objek bagian, pilih Jelajahi.
6. Arahkan ke folder yang sesuai, dan pilih satu objek untuk diunggah.
7. Pilih Buka, dan kemudian pilih Unggah.

#### Note

Jika file dengan nama yang sama sudah ada di folder yang dipilih, layanan akan menggantikan file asli dengan file yang diunggah.

Untuk mengunggah objek (AWS CLI)

- Di AWS CLI, gunakan `put-object` perintah. Anda juga dapat menyertakan salah satu parameter berikut: `content-type`, `cache-control` (untuk memungkinkan pemanggil mengontrol perilaku cache objek), dan `path` (untuk menempatkan objek dalam folder dalam wadah).

**Note**

Setelah mengunggah objek, Anda tidak dapat mengedit `content-type`, `cache-control`, atau `path`.

```
aws mediastore-data put-object --endpoint https://  
aaabbbcccddee.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
```

Contoh berikut menunjukkan nilai kembali:

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

## Melihat daftar objek

Anda dapat menggunakan AWS ElementalMediaStorekonsol untuk melihat item (objek dan folder) disimpan di tingkat paling atas wadah atau dalam folder. Item yang disimpan dalam subfolder wadah atau folder saat ini tidak akan ditampilkan. Anda dapat menggunakan AWS CLI untuk melihat daftar objek dan folder dalam wadah, terlepas dari berapa banyak folder atau subfolder dalam wadah.

Untuk melihat daftar objek di kontainer tertentu (konsol)

1. Buka MediaStorekonsol di <https://console.aws.amazon.com/mediastore/>.
2. Pada Kontainerhalaman, memilih nama kontainer yang memiliki folder yang ingin Anda lihat.
3. Pilih nama folder dari daftar.

Halaman rincian muncul, menampilkan semua folder dan objek yang disimpan dalam folder.



Untuk melihat daftar objek di folder tertentu (konsol)

1. Buka MediaStore konsol di <https://console.aws.amazon.com/mediastore/>.
2. Pada Kontainer halaman, memilih nama kontainer yang memiliki folder yang ingin Anda lihat.

Halaman rincian muncul, menampilkan semua folder dan objek yang disimpan dalam wadah.

Untuk melihat daftar objek dan folder di wadah tertentu (AWS CLI)

- Di AWS CLI, gunakan `list-items` perintah:

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

Contoh berikut menunjukkan nilai kembali:

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

#### Note

Objek yang tunduk pada `seconds_since_create` aturan tidak termasuk dalam `list-items` tanggapan.

Untuk melihat daftar objek dan folder dalam folder tertentu (AWS CLI)

- DiAWS CLI, gunakan `list-items` perintah, dengan nama folder yang ditentukan pada akhir permintaan:

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

Contoh berikut menunjukkan nilai kembali:

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

#### Note

Objek yang tunduk pada `seconds_since_create` aturan tidak termasuk dalam `list-items` tanggapan.

## Melihat detail objek

Setelah Anda mengunggah objek, AWS ElementalMediaStore menyimpan rincian seperti tanggal modifikasi, panjang konten, eTag (tag entitas), dan jenis konten. Untuk mempelajari bagaimana metadata objek digunakan, lihat [MediaStore interaksi dengan cache HTTP](#).

Untuk melihat detail objek (konsol)

1. BukaMediaStorekonsol di<https://console.aws.amazon.com/mediastore/>.
2. PadaKontainerhalaman, memilih nama kontainer yang memiliki objek yang ingin Anda lihat.
3. Jika objek yang ingin Anda lihat ada dalam folder, lanjutkan memilih nama folder sampai Anda melihat objek.
4. Pilih nama objek.

Halaman rincian muncul, menampilkan informasi tentang objek.

Untuk melihat detail objek (AWS CLI)

- DiAWS CLI, gunakan*describe-object*perintah:

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

Contoh berikut menunjukkan nilai kembali:

```
{  
  "ContentType": "image/jpeg",  
  "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",  
  "ContentLength": "2307346",  
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f5555555555555555da6d3"  
}
```

## Mengunduh objek

Anda dapat menggunakan konsol untuk mengunduh objek. Anda dapat menggunakanAWS CLIuntuk men-download objek atau hanya bagian dari sebuah objek.

Untuk mengunduh objek (konsol)

1. BukaMediaStorekonsol di<https://console.aws.amazon.com/mediastore/>.
2. PadaKontainerhalaman, memilih nama kontainer yang memiliki objek yang ingin Anda unduh.



```
"ContentType": "image/jpeg",  
"ETag": "2aa333bbcc8d8d22d777e999c88d4aa9e4dd89ff7f5555555555555555da6d3"  
}
```

## Penghapusan objek

AWS ElementalMediaStoremenawarkan pilihan yang berbeda untuk menghapus objek dari kontainer:

- [Menghapus objek individual](#). Biaya tidak berlaku.
- [Mengosongkan wadah](#) untuk menghapus semua objek dalam wadah sekaligus. Karena proses ini menggunakan panggilan API, biaya API normal berlaku.
- [Menambahkan kebijakan siklus hidup objek](#) untuk menghapus objek ketika mereka mencapai usia tertentu. Biaya tidak berlaku.

## Menghapus objek

Anda dapat menghapus objek secara terpisah menggunakan konsol atau AWS CLI. Atau, Anda dapat [menambahkan kebijakan siklus hidup objek](#) untuk secara otomatis menghapus objek setelah mereka mencapai usia tertentu dalam wadah, atau Anda dapat [mengosongkan wadah](#) untuk menghapus semua objek dalam wadah itu.

### Note

Saat Anda menghapus satu-satunya objek dalam folder, AWS ElementalMediaStoreotomatis menghapus folder dan folder kosong di atas folder itu. Misalnya, anggaplah Anda memiliki folder bernama `premium` yang tidak berisi file apa pun tetapi mengandung satu subfolder bernama `canada`. Parameter `canadasubfolder` berisi satu file bernama `law.ts`. Jika Anda menghapus file `law.ts`, layanan menghapus kedua `premium` dan `canada` folder.

Untuk menghapus objek (konsol)

1. Buka MediaStore konsol di <https://console.aws.amazon.com/mediastore/>.
2. Pada Kontainer halaman, memilih nama kontainer yang memiliki objek yang ingin Anda hapus.
3. Jika objek yang ingin Anda hapus ada di folder, lanjutkan pilih nama folder sampai Anda melihat objek.

4. Pilih opsi di sebelah kiri nama objek.
5. Pilih Delete (Hapus).

Untuk menghapus objek (AWS CLI)

- Di AWS CLI, gunakan `delete-object` perintah.

Contoh:

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

Perintah ini tidak memiliki nilai kembali.

## Mengosongkan wadah

Anda dapat mengosongkan wadah untuk menghapus semua objek yang disimpan dalam wadah. Atau, Anda dapat menambahkan [kebijakan siklus hidup objek](#) untuk secara otomatis menghapus objek setelah mereka mencapai usia tertentu dalam wadah, atau Anda dapat [menghapus objek secara individual](#).

Untuk mengosongkan kontainer (konsol)

1. Buka MediaStore konsol di <https://console.aws.amazon.com/mediastore/>.
2. Pada Kontainer halaman, memilih opsi untuk wadah yang ingin Anda kosongkan.
3. Pilih Kontainer kosong. Sebuah pesan konfirmasi akan muncul.
4. Konfirmasikan bahwa Anda ingin mengosongkan kontainer dengan memasukkan nama kontainer ke dalam kolom teks, lalu pilih Kosong.

# Keamanan di AWS Elemental MediaStore

Keamanan cloud di AWS merupakan prioritas tertinggi. Sebagai pelanggan AWS, Anda mendapatkan manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara AWS dan Anda. [Model tanggung jawab bersama](#) menggambarkan ini sebagai keamanan dari cloud dan keamanan di dalam cloud:

- Keamanan dari cloud – AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan layanan AWS di Cloud AWS Cloud. AWS juga menyediakan layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga menguji dan memverifikasi secara berkala efektivitas keamanan kami sebagai bagian dari [Program Kepatuhan AWS](#). Untuk mempelajari tentang program kepatuhan yang berlaku untuk AWS Elemental MediaStore, lihat [AWS Layanan dalam Lingkup oleh AWS Layanan Program Kepatuhan dalam Lingkup oleh Program](#).
- Keamanan di cloud – Tanggung jawab Anda ditentukan menurut layanan AWS yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain termasuk sensitivitas data Anda, persyaratan perusahaan Anda, serta hukum dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan MediaStore. Topik berikut menunjukkan cara mengonfigurasi MediaStore untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga belajar cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan MediaStore sumber daya Anda.

## Topik

- [Perlindungan data di AWS Elemental MediaStore](#)
- [Identity and Access Management untuk AWS Elemental MediaStore](#)
- [Pencatatan dan pemantauan di AWS Elemental MediaStore](#)
- [Validasi kepatuhan untuk AWS Elemental MediaStore](#)
- [Ketahanan dalam AWS Elemental MediaStore](#)
- [Keamanan Infrastruktur di AWS Elemental MediaStore](#)
- [Pencegahan confused deputy lintas layanan](#)

# Perlindungan data di AWS Elemental MediaStore

[Model tanggung jawab AWS bersama model tanggung](#) berlaku untuk perlindungan data di AWS Elemental MediaStore. Sebagaimana diuraikan dalam model ini, AWS bertanggung jawab untuk memberikan perlindungan terhadap infrastruktur global yang menjalankan semua AWS Cloud. Anda harus bertanggung jawab untuk memelihara kendali terhadap konten yang di-hosting pada infrastruktur ini. Anda juga bertanggung jawab atas tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Untuk informasi selengkapnya tentang privasi data, lihat [FAQ Privasi Data](#). Untuk informasi tentang perlindungan data di Eropa, silakan lihat postingan blog [Model Tanggung Jawab Bersama AWS dan GDPR](#) di Blog Keamanan AWS.

Untuk tujuan perlindungan data, sebaiknya Anda melindungi kredensial Akun AWS dan menyiapkan AWS IAM Identity Center atau AWS Identity and Access Management (IAM) untuk pengguna individu. Dengan cara seperti itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugas mereka. Kami juga merekomendasikan agar Anda mengamankan data Anda dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk melakukan komunikasi dengan sumber daya AWS. Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Siapkan API dan log aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi enkripsi AWS, bersama dengan semua kontrol keamanan default dalam Layanan AWS.
- Gunakan layanan keamanan terkelola lanjutan seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Untuk informasi selengkapnya tentang titik akhir FIPS yang tersedia, silakan lihat [Standar Pemrosesan Informasi Federal \(FIPS\) 140-2](#).

Sebaiknya Anda tidak memasukkan informasi rahasia atau sensitif, seperti alamat email pelanggan, ke dalam tanda atau bidang teks bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan MediaStore atau lainnya Layanan AWS menggunakan konsol, APIAWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang teks bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau diagnostik. Saat Anda memberikan URL ke server eksternal, sebaiknya jangan menyertakan informasi kredensial di URL untuk memvalidasi permintaan Anda ke server tersebut.



## Enkripsi data

MediaStore mengenkripsi wadah dan objek saat istirahat menggunakan algoritma AES-256 standar industri. Kami menyarankan Anda menggunakan MediaStore untuk mengamankan data Anda dengan cara berikut:

- Buat kebijakan kontainer untuk mengontrol hak akses ke semua folder dan objek dalam wadah itu. Untuk informasi selengkapnya, lihat [the section called “Kebijakan kontainer”](#).
- Buat kebijakan berbagi sumber daya lintas asal (CORS) untuk memungkinkan akses lintas asal secara selektif ke sumber daya Anda. MediaStore Dengan CORS, Anda dapat mengizinkan aplikasi web klien yang dimuat dalam satu domain untuk berinteraksi dengan sumber daya di domain yang berbeda. Untuk informasi selengkapnya, lihat [the section called “Kebijakan CORS”](#).

## Identity and Access Management untuk AWS Elemental MediaStore

(IAM) AWS Identity and Access Management adalah Layanan AWS yang membantu seorang administrator dalam mengendalikan akses ke sumber daya AWS secara aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya. MediaStore IAM adalah sebuah layanan Layanan AWS yang dapat Anda gunakan tanpa dikenakan biaya tambahan.

### Topik

- [Audiens](#)
- [Mengautentikasi dengan identitas](#)
- [Mengelola kebijakan menggunakan akses](#)
- [Bagaimana AWS Elemental MediaStore bekerja dengan IAM](#)
- [Contoh kebijakan berbasis identitas untuk AWS Elemental MediaStore](#)
- [Memecahkan masalah identitas dan akses AWS MediaStore Elemental](#)

## Audiens

Cara menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di MediaStore.

**Pengguna layanan** — Jika Anda menggunakan MediaStore layanan untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak MediaStore fitur untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di MediaStore, lihat [Memecahkan masalah identitas dan akses AWS MediaStore Elemental](#).

**Administrator layanan** — Jika Anda bertanggung jawab atas MediaStore sumber daya di perusahaan Anda, Anda mungkin memiliki akses penuh ke MediaStore. Tugas Anda adalah menentukan MediaStore fitur dan sumber daya mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep dasar IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM MediaStore, lihat [Bagaimana AWS Elemental MediaStore bekerja dengan IAM](#).

**Administrator IAM** – Jika Anda adalah administrator IAM, Anda mungkin ingin belajar dengan lebih detail tentang cara Anda menulis kebijakan untuk mengelola akses ke MediaStore. Untuk melihat contoh kebijakan MediaStore berbasis identitas yang dapat Anda gunakan di IAM, lihat [Contoh kebijakan berbasis identitas untuk AWS Elemental MediaStore](#)

## Mengautentikasi dengan identitas

Autentikasi merupakan cara Anda untuk masuk ke AWS dengan menggunakan kredensial identitas Anda. Anda harus terautentikasi (masuk keAWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengambil peran IAM.

Anda dapat masuk ke AWS sebagai identitas terfederasi dengan menggunakan kredensial yang disediakan melalui sumber identitas. AWS IAM Identity Center Para pengguna (Pusat Identitas IAM), otentikasi sign-on tunggal perusahaan Anda, dan kredensial Google atau Facebook Anda merupakan contoh identitas terfederasi. Saat Anda masuk sebagai identitas terfederasi, administrator Anda sebelumnya menyiapkan federasi identitas dengan menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil suatu peran.

Tergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal akses AWS. Untuk informasi selengkapnya tentang masuk ke AWS, silakan lihat [Cara masuk ke Akun AWS Anda](#) di Panduan Pengguna AWS Sign-In.

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensial Anda. Jika Anda tidak menggunakan peralatan AWS, maka Anda harus menandatangani sendiri permintaan tersebut. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, silakan lihat [Menandatangani permintaan API AWS](#) di Panduan Pengguna IAM.

Terlepas dari metode autentikasi yang Anda gunakan, Anda mungkin juga diminta untuk menyediakan informasi keamanan tambahan. Sebagai contoh, AWS menyarankan supaya Anda menggunakan autentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, silakan lihat [Autentikasi multi-faktor](#) di Panduan Pengguna AWS IAM Identity Center dan [Menggunakan autentikasi multi-faktor \(MFA\) di AWS](#) di Panduan Pengguna IAM.

## Pengguna root Akun AWS

Ketika Anda membuat Akun AWS, Anda memulai dengan satu identitas masuk yang memiliki akses ke semua Layanan AWS dan sumber daya di akun tersebut. Identitas ini disebut pengguna root Akun AWS dan diakses dengan cara masuk ke alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, silakan lihat [Tugas yang memerlukan kredensial pengguna root](#) di Panduan Pengguna IAM.

## Identitas terfederasi

Praktik terbaiknya berupa, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensial temporer.

Identitas terfederasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, dikenal sebagai AWS Directory Service, direktori Pusat Identitas, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensial yang disediakan melalui sumber identitas. Ketika identitas terfederasi mengakses Akun AWS, identitas tersebut mengambil peran, dan peran memberikan kredensial temporer.

Untuk pengelolaan akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua Akun AWS Anda dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, silakan lihat [Apakah Pusat Identitas IAM itu?](#) di User Guide AWS IAM Identity Center.

## Pengguna dan Grup IAM

[Pengguna IAM](#) adalah identitas dalam Akun AWS Anda yang memiliki izin khusus untuk satu orang atau aplikasi. Apabila memungkinkan, kami menyarankan untuk mengandalkan pada kredensial temporer alih-alih membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami menyarankan Anda memutar kunci akses. Untuk informasi selengkapnya, silakan lihat [Memutar kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang](#) di Panduan Pengguna IAM.

[Grup IAM](#) adalah identitas yang menerangkan secara spesifik kumpulan pengguna IAM. Anda tidak dapat masuk sebagai kelompok. Anda dapat menggunakan grup untuk menerangkan secara spesifik izin untuk beberapa pengguna sekaligus. Grup membuat izin lebih mudah dikelola untuk sekelompok besar pengguna. Sebagai contoh, Anda dapat memiliki grup yang diberi nama AdminIAM dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran tersebut dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial temporer. Untuk mempelajari selengkapnya, silakan lihat [Kapan harus membuat pengguna IAM \(alih-alih peran\)](#) di Panduan Pengguna IAM.

## Peran IAM

[Peran IAM](#) merupakan identitas dalam Akun AWS Anda yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Anda dapat menggunakan peran IAM untuk sementara dalam AWS Management Console dengan [berganti peran](#). Anda dapat mengambil peran dengan cara memanggil operasi API AWS CLI atau AWS atau menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, silakan lihat [menggunakan peran IAM](#) di Panduan Pengguna IAM.

IAM role dengan kredensial temporer berguna dalam situasi berikut:

- Akses pengguna gabungan – Untuk menetapkan izin ke sebuah identitas terfederasi, Anda harus membuat sebuah peran dan menentukan izin untuk peran tersebut. Ketika identitas gabungan terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberikan izin yang ditentukan oleh peran. Untuk informasi tentang peran-peran untuk federasi, silakan lihat [Membuat sebuah peran untuk Penyedia Identitas pihak ketiga](#) di Panduan Pengguna IAM. Jika Anda menggunakan Pusat Identitas IAM, Anda mengonfigurasi serangkain izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM mengkorelasikan izin yang diatur ke peran dalam IAM. Untuk informasi tentang rangkaian izin, silakan lihat [Rangkaian izin](#) di Panduan Pengguna AWS IAM Identity Center.
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM untuk sementara mengambil izin berbeda untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (pengguna utama tepercaya) di akun berbeda untuk mengakses sumber daya yang ada di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, pada beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan suatu peran sebagai proksi). Untuk mempelajari perbedaan antara kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, silakan lihat [Bagaimana peran IAM role berbeda dari kebijakan berbasis sumber daya](#) dalam Panduan Pengguna IAM.
- Akses lintas layanan – Sebagian Layanan AWS menggunakan fitur di Layanan AWS lainnya. Sebagai contoh, ketika Anda melakukan panggilan dalam suatu layanan, lazim pada layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Suatu layanan mungkin melakukan hal tersebut menggunakan izin pengguna utama panggilan, menggunakan peran layanan, atau peran tertaut layanan.
- Sesi akses maju (FAS) – Ketika Anda menggunakan pengguna IAM atau peran IAM untuk melakukan tindakan-tindakan di AWS, Anda akan dianggap sebagai seorang pengguna utama. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian dilanjutkan oleh tindakan lain pada layanan yang berbeda. FAS menggunakan izin dari pengguna utama untuk memanggil Layanan AWS, yang dikombinasikan dengan Layanan AWS yang diminta untuk membuat pengajuan ke layanan hilir. Permintaan FAS hanya diajukan ketika sebuah layanan menerima pengajuan yang memerlukan interaksi dengan Layanan AWS lain atau sumber daya lain untuk diselesaikan. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, silakan lihat [Meneruskan sesi akses](#).
- Peran layanan – Sebuah peran layanan adalah sebuah [peran IAM](#) yang dijalankan oleh suatu layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat,

memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, silakan lihat [Membuat sebuah peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.

- Peran tertaut layanan – Peran tertaut layanan adalah tipe peran layanan yang tertaut dengan Layanan AWS. Layanan tersebut dapat menjalankan peran untuk melakukan sebuah tindakan atas nama Anda. Peran tertaut layanan akan muncul di Akun AWS Anda dan dimiliki oleh layanan tersebut. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran tertaut layanan.
- Aplikasi yang berjalan di Amazon EC2 – Anda dapat menggunakan peran IAM untuk mengelola kredensial temporer untuk aplikasi yang berjalan di instans EC2 dan mengajukan permintaan AWS CLI atau API AWS. Cara ini lebih baik daripada menyimpan kunci akses dalam instans EC2. Untuk menugaskan sebuah peran AWS ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda dapat membuat sebuah profil instans yang dilampirkan ke instans. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 untuk mendapatkan kredensial sementara. Untuk informasi selengkapnya, silakan lihat [Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di instans Amazon EC2](#) di Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, silakan lihat [Kapan harus membuat peran IAM \(alih-alih pengguna\)](#) di Panduan Pengguna IAM.

## Mengelola kebijakan menggunakan akses

Anda mengendalikan akses di AWS dengan membuat kebijakan dan melampirkannya ke identitas atau sumber daya AWS. Kebijakan adalah objek di AWS yang, ketika terkait dengan identitas atau sumber daya, akan menentukan izinnya. AWS mengevaluasi kebijakan-kebijakan tersebut ketika seorang pengguna utama (pengguna, root user, atau sesi peran) mengajukan permintaan. Izin dalam kebijakan menentukan apakah permintaan diberikan atau ditolak. Sebagian besar kebijakan disimpan di AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, silakan lihat [Gambaran Umum kebijakan JSON](#) di Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses pada apa. Yaitu, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan syarat apa.

Secara bawaan, para pengguna dan peran tidak memiliki izin. Untuk mengabulkan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat

membuat kebijakan IAM. Administrator kemudian akan dapat menambahkan kebijakan IAM ke peran, dan para pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk pengoperasiannya. Sebagai contoh, anggap saja Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut dapat memperoleh informasi peran dari AWS Management Console, AWS CLI, atau APIAWS.

## Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, misalnya pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol apa yang pengguna tindakan dan peran dapat kerjakan, pada sumber daya mana, dan dalam keadaan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, silakan lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan terkelola. Kebijakan inline ditanam secara langsung ke pengguna tunggal, grup, atau peran. Kebijakan terkelola adalah kebijakan yang berdiri sendiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran di Akun AWS Anda. Kebijakan terkelola mencakup kebijakan terkelola AWS dan kebijakan terkelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan inline, silakan lihat [Memilih antara kebijakan terkelola dan kebijakan inline](#) di Panduan Pengguna IAM.

## Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan-kebijakan berbasis sumber daya adalah kebijakan terpercaya peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya yang dilampiri kebijakan, kebijakan tersebut menentukan tindakan apa yang dapat dilakukan oleh pengguna utama yang ditentukan di sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan pengguna utama](#) dalam kebijakan berbasis sumber daya. Pengguna utama dapat mencakup akun, pengguna, peran, pengguna gabungan, atau Layanan AWS.

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan terkelola AWS dari IAM dalam kebijakan berbasis sumber daya.

## Daftar kontrol akses (ACL)

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan-kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh-contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, silakan lihat [Gambaran umum daftar kontrol akses \(ACL\)](#) di Panduan Pengembang Layanan Penyimpanan Ringkas Amazon.

## Tipe-tipe kebijakan lain

AWS mendukung tipe kebijakan tambahan, yang kurang umum. Tipe-tipe kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh tipe kebijakan yang lebih umum.

- **Batasan izin** – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM (pengguna IAM atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batas izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini menindahi izin. Untuk informasi selengkapnya tentang batasan izin, silakan lihat [Batasan izin untuk entitas IAM](#) di Panduan Pengguna IAM.
- **Kebijakan kontrol layanan (SCP)** – SCP adalah kebijakan JSON yang menentukan izin maksimum untuk sebuah organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations adalah layanan untuk mengelompokkan dan secara terpusat mengelola beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur di suatu organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau ke semua akun Anda. SCP membatasi izin untuk entitas dalam akun anggota, termasuk setiap Pengguna root akun AWS. Untuk informasi selengkapnya tentang Organisasi dan SCP, silakan lihat [Cara kerja SCP](#) di Panduan Pengguna AWS Organizations.
- **Kebijakan sesi** – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara terprogram untuk peran atau pengguna gabungan. Izin sesi yang dihasilkan adalah perpotongan kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga dapat berasal dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini menindahi izin. Untuk informasi selengkapnya, silakan lihat [Kebijakan sesi](#) di Panduan Pengguna IAM.



## Berbagai tipe kebijakan

Ketika beberapa tipe kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan ketika beberapa tipe kebijakan dilibatkan, silakan lihat [Logika evaluasi kebijakan](#) di Panduan Pengguna IAM.

## Bagaimana AWS Elemental MediaStore bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses MediaStore, pelajari fitur IAM yang tersedia untuk digunakan. MediaStore

Fitur IAM yang dapat Anda gunakan dengan AWS Elemental MediaStore

Fitur IAM	MediaStore dukungan
<a href="#">Kebijakan berbasis identitas</a>	Ya
<a href="#">Kebijakan berbasis sumber daya</a>	Ya
<a href="#">Tindakan kebijakan</a>	Ya
<a href="#">Sumber daya kebijakan</a>	Ya
<a href="#">kunci-kunci persyaratan kebijakan (spesifik layanan)</a>	Ya
<a href="#">ACL</a>	Tidak
<a href="#">ABAC (tag dalam kebijakan)</a>	Parsial
<a href="#">Kredensial temporer</a>	Ya
<a href="#">Izin-izin pengguna utama</a>	Ya
<a href="#">Peran layanan</a>	Ya
<a href="#">Peran tertaut layanan</a>	Tidak

Untuk mendapatkan tampilan tingkat tinggi tentang cara MediaStore dan AWS layanan lain bekerja dengan sebagian besar fitur IAM, lihat [AWSlayanan yang bekerja dengan IAM di Panduan Pengguna IAM](#).

## Kebijakan berbasis identitas untuk MediaStore

Mendukung kebijakan berbasis identitas Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, misalnya pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol apa yang pengguna tindakan dan peran dapat kerjakan, pada sumber daya mana, dan dalam keadaan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, silakan lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta persyaratan yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Anda tidak dapat menentukan secara spesifik pengguna utama dalam sebuah kebijakan berbasis identitas karena pengguna utama berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, silakan lihat [Referensi elemen kebijakan JSON IAM](#) dalam Panduan Pengguna IAM.

### Contoh kebijakan berbasis identitas untuk MediaStore

Untuk melihat contoh kebijakan MediaStore berbasis identitas, lihat. [Contoh kebijakan berbasis identitas untuk AWS Elemental MediaStore](#)

## Kebijakan berbasis sumber daya dalam MediaStore

Mendukung kebijakan berbasis sumber daya Ya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan-kebijakan berbasis sumber daya adalah kebijakan terpercaya peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya yang dilampiri kebijakan, kebijakan tersebut menentukan tindakan apa yang dapat

dilakukan oleh pengguna utama yang ditentukan di sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan pengguna utama](#) dalam kebijakan berbasis sumber daya. Pengguna utama dapat mencakup akun, pengguna, peran, pengguna gabungan, atau Layanan AWS.

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai pengguna utama dalam kebijakan berbasis sumber daya. Menambahkan pengguna utama akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan. Ketika pengguna utama dan sumber daya berada dalam Akun AWS yang berbeda, Administrator IAM di akun tepercaya juga harus memberikan izin kepada entitas pengguna utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses kepada pengguna utama dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, silakan lihat [Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya](#) di Panduan Pengguna IAM.

#### Note

MediaStore juga mendukung kebijakan kontainer yang menentukan entitas utama mana (akun, pengguna, peran, dan pengguna federasi) yang dapat melakukan tindakan pada penampung. Untuk informasi selengkapnya, lihat [Kebijakan kontainer](#).

## Tindakan kebijakan untuk MediaStore

Mendukung tindakan kebijakan

Ya

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses pada apa. Yaitu, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan syarat apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan-tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan-tindakan kebijakan biasanya memiliki nama yang sama sebagaimana operasi API AWS yang dikaitkan padanya. Ada beberapa pengecualian, misalnya tindakan yang memiliki izin saja yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam sebuah kebijakan. Tindakan-tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam suatu kebijakan untuk memberikan izin guna melakukan operasi yang terkait.

Untuk melihat daftar MediaStore tindakan, lihat [Tindakan yang ditentukan oleh AWS Elemental MediaStore](#) dalam Referensi Otorisasi Layanan.

Tindakan kebijakan MediaStore menggunakan awalan berikut sebelum tindakan:

```
mediastore
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan-tindakan tersebut dengan koma.

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

Untuk melihat contoh kebijakan MediaStore berbasis identitas, lihat [Contoh kebijakan berbasis identitas untuk AWS Elemental MediaStore](#)

## Sumber daya kebijakan untuk MediaStore

Mendukung sumber daya kebijakan	Ya
---------------------------------	----

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses pada apa. Yaitu, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan syarat apa.

Elemen kebijakan JSON Resource menentukan objek atau objek-objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan entah elemen Resource atau NotResource. Praktik terbaiknya, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Anda dapat melakukan ini untuk tindakan-tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (\*) untuk mengindikasikan bahwa pernyataan tersebut berlaku bagi semua sumber daya.

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

Untuk melihat daftar jenis MediaStore sumber daya dan ARNnya, lihat Sumber [daya yang ditentukan oleh AWS MediaStore](#) Elemental dalam Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat [Tindakan yang ditentukan oleh AWS Elemental](#). MediaStore

Sumber daya MediaStore kontainer memiliki ARN berikut:

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

Untuk informasi lebih lanjut tentang format ARN, lihat [Amazon Resource Name \(ARN\) dan Namespace Layanan AWS](#).

Misalnya, untuk menentukan AwardsShow wadah dalam pernyataan Anda, gunakan ARN berikut:

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

## Kunci kondisi kebijakan untuk MediaStore

Mendukung kunci-kunci persyaratan kebijakan spesifik layanan	Ya
--	----

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses pada apa. Yaitu, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan syarat apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan syarat yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan [operator syarat](#), misalnya sama dengan atau kurang dari, untuk mencocokkan syarat dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya dengan menggunakan operasi AND yang logis. Jika Anda menentukan beberapa nilai untuk satu kunci persyaratan, maka AWS akan mengevaluasi syarat tersebut menggunakan operasi OR yang logis. Semua persyaratan harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan syarat. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tag yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, silakan lihat [Elemen kebijakan IAM: variabel dan tag](#) di Panduan Pengguna IAM.

AWS mendukung kunci-kunci syarat global dan kunci-kunci syarat spesifik layanan. Untuk melihat semua kunci persyaratan global AWS, silakan lihat [kunci konteks syarat global AWS](#) di Panduan Pengguna IAM.

Untuk melihat daftar kunci kondisi, lihat Kunci MediaStore kondisi [untuk AWS Elemental MediaStore](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat [Tindakan yang ditentukan oleh AWS Elemental MediaStore](#).

Untuk melihat contoh kebijakan MediaStore berbasis identitas, lihat. [Contoh kebijakan berbasis identitas untuk AWS Elemental MediaStore](#)

## ACL di MediaStore

Mendukung ACL

Tidak

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan-kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

## ABAC dengan MediaStore

Mendukung ABAC (tag dalam kebijakan)

Parsial

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Di AWS, atribut-atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna

atau peran) dan ke banyak sumber daya AWS. Pemberian tag ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi-operasi ketika tag milik pengguna utama cocok dengan tag yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi dimana pengelolaan kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tag di [elemen syarat](#) dari sebuah kebijakan dengan menggunakan kunci-kunci persyaratan `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci-kunci persyaratan untuk setiap jenis sumber daya, maka nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci persyaratan untuk hanya beberapa jenis sumber daya, maka nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, silakan lihat [Apa itu ABAC?](#) di Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, silakan lihat [Menggunakan kontrol akses berbasis atribut \(ABAC\)](#) di Panduan Pengguna IAM.

## Menggunakan kredensial sementara dengan MediaStore

Mendukung kredensial temporer

Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk dengan menggunakan kredensial temporer. Sebagai informasi tambahan, termasuk tentang Layanan AWS mana saja yang berfungsi dengan kredensial temporer, silakan lihat [Layanan AWS yang berfungsi dengan IAM](#) di Panduan Pengguna IAM.

Anda menggunakan kredensial temporer jika Anda masuk ke AWS Management Console dengan menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Sebagai contoh, ketika Anda mengakses AWS dengan menggunakan tautan masuk tunggal (SSO) milik perusahaan Anda, proses itu secara otomatis akan membuat kredensial temporer. Anda juga akan secara otomatis membuat kredensial temporer ketika Anda masuk ke konsol sebagai seorang pengguna dan kemudian beralih peran. Untuk informasi selengkapnya tentang peralihan peran, silakan lihat [Peralihan peran \(konsol\)](#) di Panduan Pengguna IAM.

Anda dapat secara manual membuat kredensial temporer menggunakan AWS CLI atau API AWS. Anda kemudian dapat menggunakan kredensial temporer tersebut untuk mengakses AWS. AWS

menyarankan agar Anda secara dinamis membuat kredensial temporer alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, silakan lihat [Kredensial keamanan temporer di IAM](#).

## Izin utama lintas layanan untuk MediaStore

Mendukung sesi akses maju (FAS)	Ya
---------------------------------	----

Saat Anda menggunakan pengguna IAM atau peran IAM untuk mengerjakan tindakan di AWS, Anda akan dianggap sebagai pengguna utama. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian dilanjutkan oleh tindakan lain pada layanan yang berbeda. FAS menggunakan izin dari pengguna utama untuk memanggil Layanan AWS, yang dikombinasikan dengan Layanan AWS yang diminta untuk membuat pengajuan ke layanan hilir. Permintaan FAS hanya diajukan ketika sebuah layanan menerima pengajuan yang memerlukan interaksi dengan Layanan AWS lain atau sumber daya lain untuk diselesaikan. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, silakan lihat [Meneruskan sesi akses](#).

## Peran layanan untuk MediaStore

Mendukung peran layanan	Ya
-------------------------	----

Peran layanan adalah sebuah [peran IAM](#) yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.

### Warning

Mengubah izin untuk peran layanan dapat merusak MediaStore fungsionalitas. Edit peran layanan hanya jika MediaStore memberikan panduan untuk melakukannya.



## Peran terkait layanan untuk MediaStore

Mendukung peran yang tertaut dengan layanan    Tidak

Peran yang tertaut layanan adalah jenis peran layanan yang tertaut dengan Layanan AWS. Layanan tersebut dapat menjalankan peran untuk melakukan sebuah tindakan atas nama Anda. Peran tertaut layanan akan muncul di Akun AWS Anda dan dimiliki oleh layanan tersebut. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran tertaut layanan.

Untuk detail tentang pembuatan atau pengelolaan peran yang terhubung dengan layanan, lihat [Layanan AWS yang bekerja dengan IAM](#). Cari layanan dalam tabel yang memiliki Yes di kolom Service-linked role (Peran yang terhubung dengan layanan). Pilih tautan Ya untuk melihat dokumentasi peran tertaut layanan untuk layanan tersebut.

## Contoh kebijakan berbasis identitas untuk AWS Elemental MediaStore

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi MediaStore sumber daya. Pengguna dan peran tersebut juga tidak dapat melakukan tugas dengan menggunakan API AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS. Untuk mengabdikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian akan dapat menambahkan kebijakan IAM ke peran, dan para pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, silakan lihat [Membuat kebijakan IAM](#) di Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh MediaStore, termasuk format ARN untuk setiap jenis sumber daya, lihat [Kunci tindakan, sumber daya, dan kondisi untuk AWS MediaStore](#) Elemental dalam Referensi Otorisasi Layanan.

### Topik

- [Praktik terbaik kebijakan](#)
- [Menggunakan konsol MediaStore](#)
- [Perbolehkan pengguna untuk melihat izin mereka sendiri](#)

## Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus MediaStore sumber daya di akun Anda. Tindakan ini mengenakan biaya kepada Anda Akun AWS. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan terkelola AWS dan beralih ke izin dengan hak akses paling rendah – Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan terkelola AWS yang memberikan izin untuk banyak kasus penggunaan umum. Kebijakan terdapat di Akun AWS Anda. Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan AWS yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, silakan lihat [kebijakan-kebijakan terkelola AWS](#) atau [kebijakan-kebijakan terkelola AWS untuk fungsi tugas](#) di Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukan ini dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan pengguna IAM untuk mengajukan izin, silakan lihat [Kebijakan dan izin di IAM](#) di Panduan Pengguna IAM.
- Gunakan syarat dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu syarat ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis syarat kebijakan untuk menentukan bahwa semua pengajuan harus dikirim menggunakan SSL. Anda juga dapat menggunakan syarat untuk memberi akses ke tindakan layanan jika digunakan melalui Layanan AWS yang spesifik, seperti AWS CloudFormation. Untuk informasi selengkapnya, silakan lihat [Elemen kebijakan JSON IAM: Syarat](#) di Panduan Pengguna IAM.
- Gunakan Analizer Akses IAM untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – Analizer Akses IAM memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. Analizer Akses IAM menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, silakan lihat [validasi kebijakan Analizer Akses IAM](#) di Panduan Pengguna IAM.
- Memerlukan autentikasi multi-faktor (MFA) – Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Akun AWS Anda, aktifkan MFA untuk keamanan tambahan.

Untuk meminta MFA ketika operasi API dipanggil, tambahkan syarat MFA pada kebijakan Anda. Untuk informasi selengkapnya, silakan lihat [Mengonfigurasi akses API yang diproteksi MFA](#) di Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, silakan lihat [Praktik terbaik keamanan di IAM](#) di Panduan Pengguna IAM.

## Menggunakan konsol MediaStore

Untuk mengakses MediaStore konsol AWS Elemental, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang MediaStore sumber daya di AndaAkun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu meloloskan izin konsol minimum bagi pengguna yang hanya melakukan panggilan ke API AWS CLI atau AWS. Jika tidak, akses hanya diizinkan ke tindakan-tindakan yang sesuai dengan operasi API yang sedang mereka coba lakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan MediaStore konsol, lampirkan juga kebijakan MediaStore *ConsoleAccess* atau *ReadOnly* AWS terkelola ke entitas. Untuk informasi selengkapnya, lihat [Menambahkan izin ke pengguna](#) dalam Panduan Pengguna IAM.

## Perbolehkan pengguna untuk melihat izin mereka sendiri

Contoh ini menunjukkan cara Anda dapat membuat kebijakan yang mengizinkan para pengguna IAM untuk melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan pada konsol atau secara terprogram menggunakan API AWS CLI atau 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": "*"
}
]
```

## Memecahkan masalah identitas dan akses AWS MediaStore Elemental

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan MediaStore dan IAM.

### Topik

- [Saya tidak berwenang untuk melakukan tindakan di MediaStore](#)
- [Saya tidak berwenang untuk melakukan iam: PassRole](#)
- [Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses MediaStore sumber daya saya](#)

### Saya tidak berwenang untuk melakukan tindakan di MediaStore

Jika Anda menerima pesan galat bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh galat berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya `my-example-widget` rekaan, tetapi tidak memiliki izin `mediastore:GetWidget` rekaan.

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

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya `my-example-widget` dengan menggunakan tindakan `mediastore:GetWidget`.

Jika Anda membutuhkan bantuan, hubungi administrator AWS Anda. Administrator Anda adalah orang yang memberikan kredensial masuk Anda.

## Saya tidak berwenang untuk melakukan `iam:PassRole`

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran MediaStore.

Sebagian Layanan AWS mengizinkan Anda untuk memberikan peran yang sudah ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran tertaut-layanan. Untuk melakukan tindakan tersebut, Anda harus memiliki izin untuk memberikan peran pada layanan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol tersebut untuk melakukan tindakan di MediaStore. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

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

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda membutuhkan bantuan, hubungi administrator AWS Anda. Administrator Anda adalah orang yang memberikan kredensial masuk Anda.

## Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses MediaStore sumber daya saya

Anda dapat membuat peran yang dapat digunakan para pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa yang

dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi akses kepada orang ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa hal berikut:

- Untuk mempelajari apakah MediaStore mendukung fitur-fitur ini, lihat [Bagaimana AWS Elemental MediaStore bekerja dengan IAM](#).
- Untuk mempelajari cara memberikan akses ke sumber daya di seluruh Akun AWS yang Anda miliki, silakan lihat [Menyediakan akses ke pengguna IAM di Akun AWS lainnya yang Anda miliki](#) di Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses ke sumber daya Anda ke pihak ketiga Akun AWS, silakan lihat [Menyediakan akses ke akun Akun AWS yang dimiliki oleh pihak ketiga](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, silakan lihat [Menyediakan akses ke pengguna terautentikasi eksternal \(gabungan identitas\)](#) di Panduan Pengguna IAM .
- Untuk mempelajari perbedaan antara penggunaan kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, silakan lihat [Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya](#) di Panduan Pengguna IAM.

## Pencatatan dan pemantauan di AWS Elemental MediaStore

Bagian ini memberikan gambaran umum tentang opsi untuk masuk dan memantau AWS Elemental MediaStore untuk tujuan keamanan. Untuk informasi selengkapnya tentang pencatatan dan pemantauan MediaStore, lihat [Pemantauan dan penandaan di AWS Elemental MediaStore](#).

Pemantauan adalah bagian penting dari pemeliharaan keandalan, ketersediaan, dan performa AWS Elemental MediaStore serta solusi AWS Anda. Anda harus mengumpulkan data pemantauan dari semua bagian AWS solusi Anda sehingga Anda dapat lebih mudah men-debug kegagalan multi-titik jika terjadi. AWS menyediakan beberapa alat untuk memantau MediaStore sumber daya Anda dan menanggapi potensi insiden.

### CloudWatch Alarm Amazon

Menggunakan CloudWatch alarm, Anda menonton satu metrik selama periode waktu yang Anda tentukan. Jika metrik melebihi ambang batas tertentu, notifikasi akan dikirim ke topik Amazon SNS atau kebijakan AWS Auto Scaling. CloudWatch alarm tidak memanggil tindakan karena mereka

berada dalam keadaan tertentu. Sebaliknya, negara harus telah berubah dan dipertahankan untuk sejumlah periode tertentu. Untuk informasi selengkapnya, lihat [Pemantauan CloudWatch dengan](#).

## Log AWS CloudTrail

CloudTrail menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di AWS Elemental MediaStore. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat MediaStore, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan. Untuk informasi selengkapnya, lihat [Pencatatan log panggilan API log dengan CloudTrail](#).

## AWS Trusted Advisor

Trusted Advisor mengacu pada praktik terbaik yang dipelajari dari melayani ratusan ribu AWS pelanggan. Trusted Advisor memeriksa lingkungan AWS Anda dan kemudian membuat rekomendasi ketika ada peluang untuk menghemat uang, meningkatkan ketersediaan dan kinerja sistem, atau membantu menutup celah keamanan. Semua AWS pelanggan memiliki akses ke lima cek Trusted Advisor. Pelanggan dengan rencana Business atau dukungan Korporasi dapat melihat semua pemeriksaan Trusted Advisor.

Untuk informasi selengkapnya, lihat [AWS Trusted Advisor](#).

## Validasi kepatuhan untuk AWS Elemental MediaStore


Untuk mempelajari apakah Layanan AWS berada dalam lingkup program kepatuhan khusus, lihat [Layanan AWS di Scope oleh Program](#) Program Kepatuhan yang Anda minati. Untuk informasi umum, silakan lihat [Program Kepatuhan AWS](#).

Anda bisa mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh Laporan di AWS Artifact](#).

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, serta hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Panduan Quick Start Keamanan dan Kepatuhan](#) – Panduan Quick Start Keamanan dan Kepatuhan – Panduan deployment ini membahas pertimbangan arsitektur dan menyediakan langkah-langkah untuk melakukan deployment terhadap lingkungan dasar di AWS yang menjadi fokus keamanan dan kepatuhan.

- [Merancang Keamanan dan Kepatuhan HIPAA di Amazon Web Services](#) – Laporan resmi ini menjelaskan cara perusahaan dapat menggunakan AWS untuk membuat aplikasi yang memenuhi syarat HIPAA.

 Note

Tidak semua Layanan AWS memenuhi syarat HIPAA. Untuk informasi selengkapnya, lihat [Referensi Layanan yang Memenuhi Syarat HIPAA](#).

- [Sumber Daya Kepatuhan AWS](#) – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [Panduan Kepatuhan Pelanggan AWS](#) – Pahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan kontrol keamanan di banyak kerangka kerja (termasuk National Institute of Standards and Technology (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- [Mengevaluasi Sumber Daya dengan Aturan](#) di Panduan Developer AWS Config – Layanan AWS Config menilai seberapa baik konfigurasi sumber daya Anda dalam mematuhi praktik-praktik internal, pedoman industri, dan regulasi internal.
- [AWS Security Hub](#) – Layanan AWS ini memberikan pandangan komprehensif tentang status keamanan Anda di dalam AWS. Security Hub menggunakan kontrol keamanan untuk sumber daya AWS Anda dan untuk memeriksa kepatuhan terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat [Referensi kontrol Security Hub](#).
- [AWS Audit Manager](#) – Layanan AWS ini akan membantu Anda untuk terus-menerus mengaudit penggunaan AWS untuk menyederhanakan bagaimana Anda mengelola risiko dan kepatuhan terhadap regulasi dan standar industri.

## Ketahanan dalam AWS Elemental MediaStore

Infrastruktur global AWS dibangun di sekitar Wilayah AWS dan Availability Zone. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi yang terhubung dengan jaringan latensi rendah, throughput tinggi, dan jaringan yang sangat berlebihan. Dengan Availability Zone, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis mengalami fail over antar zona tanpa gangguan. Availability Zone memiliki ketersediaan yang lebih baik, toleran terhadap kegagalan, dan dapat diukur skalanya jika dibandingkan dengan satu atau beberapa infrastruktur pusat data tradisional.



Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat [Infrastruktur Global AWS](#).

Selain infrastruktur AWS global, MediaStore menawarkan beberapa fitur untuk membantu mendukung ketahanan data dan kebutuhan cadangan Anda.

## Keamanan Infrastruktur di AWS Elemental MediaStore

Sebagai layanan terkelola, AWS Elemental MediaStore dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan keamanan AWS dan cara AWS melindungi infrastruktur, lihat [Keamanan Cloud AWS](#). Guna mendesain lingkungan AWS Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat [Perlindungan Infrastruktur](#) dalam Kerangka Kerja AWS Well-Architected Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses MediaStore melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Cipher cocok dengan perfect forward secrecy (PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan sistem yang lebih baru mendukung mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan pengguna utama IAM. Atau Anda bisa menggunakan [AWS Security Token Service](#) (AWS STS) untuk membuat kredensial keamanan sementara guna menandatangani permintaan.

## Pencegahan confused deputy lintas layanan

Masalah confused deputy adalah masalah keamanan saat entitas yang tidak memiliki izin untuk melakukan suatu tindakan dapat memaksa entitas yang lebih berhak untuk melakukan tindakan tersebut. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan panggilan) memanggil layanan lain (layanan yang disebut). Layanan panggilan dapat dimanipulasi untuk menggunakan izinnya untuk bertindak atas sumber daya pelanggan lain dengan cara yang seharusnya tidak memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS sediakan alat yang

membantu Anda melindungi data Anda untuk semua layanan dengan prinsip layanan yang telah diberikan akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan global dalam kebijakan sumber daya untuk membatasi izin yang diberikan AWS MediaStore Elemental layanan lain ke sumber daya. Gunakan `aws:SourceArn` jika Anda hanya ingin satu sumber daya dikaitkan dengan akses lintas layanan. Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun itu dikaitkan dengan penggunaan lintas layanan.

Cara paling efektif untuk melindungi dari masalah wakil yang membingungkan adalah dengan menggunakan kunci konteks kondisi `aws:SourceArn` global dengan ARN penuh sumber daya. Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan kunci konteks kondisi `aws:SourceArn` global dengan karakter wildcard (\*) untuk bagian ARN yang tidak diketahui. Sebagai contoh, `arn:aws:service:*:123456789012:*`.

Jika `aws:SourceArn` nilainya tidak berisi ID akun, seperti ARN bucket Amazon S3, Anda harus menggunakan kedua kunci konteks kondisi global untuk membatasi izin.

Nilai `aws:SourceArn` harus berupa konfigurasi yang MediaStore menerbitkan CloudWatch log di Wilayah dan akun Anda.

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan MediaStore untuk mencegah masalah wakil yang membingungkan.

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

# Pemantauan dan penandaan di AWS Elemental MediaStore

Pemantauan adalah bagian penting dari pemeliharaan keandalan, ketersediaan, dan performa AWS Elemental MediaStore dan AWS solusi Anda lainnya. AWS menyediakan alat pemantauan berikut untuk mengawasi MediaStore, melaporkan saat terjadi kesalahan, dan mengambil tindakan otomatis jika diperlukan:

- AWS CloudTrail merekam panggilan API dan peristiwa terkait yang dilakukan oleh atau atas nama akun AWS Anda dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang memanggil AWS, alamat IP sumber yang melakukan panggilan, dan kapan panggilan tersebut terjadi. Untuk mengetahui informasi selengkapnya, lihat [Panduan Pengguna AWS CloudTrail](#).
- Amazon CloudWatch memantau AWS sumber daya Anda dan aplikasi yang Anda jalankan AWS di secara waktu nyata. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas yang ditentukan. Misalnya, Anda dapat membuat CloudWatch melacak penggunaan CPU atau metrik lain dari instans Amazon EC2 Anda dan secara otomatis meluncurkan instans baru ketika diperlukan. Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#).
- Amazon CloudWatch Events memberikan aliran peristiwa sistem yang menjelaskan perubahan dalam AWS sumber daya. Biasanya, AWS layanan mengirimkan pemberitahuan peristiwa ke CloudWatch Acara dalam hitungan detik, tetapi terkadang perlu waktu satu menit atau lebih. CloudWatch Peristiwa memungkinkan komputasi berbasis peristiwa otomatis, karena Anda dapat menulis aturan yang mengawasi peristiwa tertentu dan memicu tindakan otomatis di AWS layanan lainnya saat peristiwa ini terjadi. Untuk informasi selengkapnya, lihat [Panduan Pengguna Amazon CloudWatch Events](#).
- Amazon CloudWatch Logs memungkinkan Anda memantau, menyimpan, dan mengakses file log dari instans Amazon EC2, CloudTrail, dan sumber lainnya. CloudWatch Log dapat memantau informasi dalam file log dan memberi tahu Anda ketika ambang tertentu terpenuhi. Anda juga dapat mengarsipkan data log Anda dalam penyimpanan yang sangat tahan lama. Untuk informasi selengkapnya, lihat [Panduan Pengguna Amazon CloudWatch Logs](#).

Anda juga dapat menetapkan metadata ke MediaStore kontainer Anda dalam bentuk tag. Masing-masing tag adalah sebuah label yang terdiri dari sebuah kunci dan nilai yang Anda tentukan. Tag memudahkan untuk mengelola, pencarian, dan mem-filter sumber daya. Anda dapat menggunakan

tag untuk mengatur AWS sumber daya Anda di AWS Management Console, membuat laporan penggunaan dan penagihan di semua AWS sumber daya Anda, dan memfilter sumber daya selama aktivitas otomatisasi infrastruktur.

#### Topik

- [Membuat log panggilan MediaStore API AWS Elemental dengan AWS CloudTrail](#)
- [Memantau AWS Elemental MediaStore dengan Amazon CloudWatch](#)
- [Menandai sumber daya MediaStore AWS Elemental](#)

## Membuat log panggilan MediaStore API AWS Elemental dengan AWS CloudTrail

AWS Elemental MediaStore terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di MediaStore. CloudTrail menangkap subset panggilan API untuk MediaStore sebagai tindakan, termasuk panggilan dari MediaStore konsol tersebut dan dari panggilan MediaStore API. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman tindakan CloudTrail berkelanjutan ke bucket Amazon S3, termasuk peristiwa untuk MediaStore. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol di Riwayat peristiwa. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke MediaStore, alamat IP asal permintaan tersebut dibuat, siapa yang membuat permintaan, kapan permintaan dibuat ke, dan lainnya.

Untuk mempelajari selengkapnya CloudTrail, termasuk cara mengonfigurasi dan mengaktifkannya, lihat [Panduan AWS CloudTrail Pengguna](#).

#### Topik

- [MediaStore Informasi AWS Elemental di CloudTrail](#)
- [Contoh: Entri file MediaStore log AWS Elemental](#)

## MediaStore Informasi AWS Elemental di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Saat aktivitas peristiwa yang didukung terjadi di AWS Elemental MediaStore, aktivitas tersebut dicatat di CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya di Riwayat peristiwa. Anda dapat melihat, mencari,

dan mengunduh peristiwa terbaru di akun AWS Anda. Untuk informasi selengkapnya, lihat [Melihat Kejadian dengan Riwayat CloudTrail Kejadian](#).

Untuk catatan berkelanjutan tentang peristiwa di akun AWS Anda, termasuk peristiwa untuk MediaStore, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan berkas log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di dalam konsol tersebut, jejak diterapkan ke semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di partisi AWS dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lainnya untuk menganalisis lebih lanjut dan bertindak berdasarkan data peristiwa yang dikumpulkan di CloudTrail log. Untuk informasi lain, lihat topik berikut:

- [Ikhtisar untuk Membuat Jejak](#)
- [CloudTrail Layanan dan Integrasi yang Didukung](#)
- [Mengonfigurasi Notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima Berkas CloudTrail Log dari Beberapa Wilayah](#) dan [Menerima Berkas CloudTrail Log dari Beberapa Akun](#)

AWS Elemental MediaStore mendukung pencatatan operasi berikut sebagai peristiwa dalam file CloudTrail log:

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

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut ini:

- Jika permintaan tersebut dibuat dengan kredensial pengguna root atau kredensial pengguna

- Jika permintaan tersebut dibuat dengan kredensial keamanan sementara untuk peran atau pengguna gabungan
- Jika permintaan tersebut dibuat oleh layanan AWS lainnya

Untuk informasi lain, lihat [Elemen userIdentity CloudTrail](#).

## Contoh: Entri file MediaStore log AWS Elemental

Jejak adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai berkas log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau beberapa entri log berisi satu atau beberapa entri log. Peristiwa mewakili satu permintaan dari sumber apa pun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail berkas log bukan jejak tumpukan panggilan API publik yang berurutan, sehingga berkas log tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `CreateContainer` operasi:

```
{
  "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":
  "MNCTGH4HRQJ27GRMBVDPIVHEP4L02BN6MUVHBCPSHOAWNS0KSXC024B2UE0BBND5DONRXTMFK3TOJ4G7AHWMESI",
  "eventID": "7085b140-fb2c-409b-a329-f567912d704c",
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}

```

## Memantau AWS Elemental MediaStore dengan Amazon CloudWatch

Anda dapat memantau MediaStore penggunaan AWS Elemental CloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik yang dapat dibaca. CloudWatch menyimpan statistik selama 15 bulan, sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang performa aplikasi web atau layanan Anda. Anda juga dapat mengatur alarm yang memperhatikan ambang batas tertentu dan mengirim notifikasi atau mengambil tindakan saat ambang batas tersebut terpenuhi. Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#).

AWS menyediakan alat pemantauan berikut untuk mengawasi MediaStore, melaporkan saat terjadi kesalahan, dan mengambil tindakan otomatis jika diperlukan:

- Amazon CloudWatch Logs memungkinkan Anda untuk memantau, menyimpan, dan mengakses file log dari AWS layanan seperti AWS Elemental MediaStore. Anda dapat menggunakan CloudWatch Log untuk memantau aplikasi dan sistem menggunakan data log. Misalnya, CloudWatch Log dapat melacak jumlah kesalahan yang terjadi di log aplikasi dan mengirimkan notifikasi kapan pun tingkat kesalahan melebihi ambang batas yang Anda tentukan. CloudWatch Log menggunakan data log untuk pemantauan, jadi tidak ada perubahan kode yang diperlukan. Misalnya, Anda dapat memantau log aplikasi untuk istilah literal tertentu (seperti



"ValidationException") atau menghitung jumlah permintaan yang dibuat untuk istilah literal tertentu (seperti "") atau menghitung jumlah permintaan yang dibuat selama periode literal tertentu (seperti "") atau menghitung jumlahPutObject permintaan yang dibuat selama periode literal tertentu (seperti "") atau menghitung jumlah permintaan yang dibuat untuk jangka. Ketika istilah yang dicari ditemukan, CloudWatch Log akan melaporkan data ke CloudWatch metrik yang Anda tentukan. Data log dienkripsi saat transit dan saat diam.

- Amazon CloudWatch Events memberikan peristiwa sistem yang menjelaskan perubahan dalam AWS sumber daya, seperti MediaStore objek. Biasanya, AWS layanan mengirimkan pemberitahuan peristiwa ke CloudWatch Acara dalam hitungan detik, tetapi terkadang perlu waktu satu menit atau lebih. Anda dapat menyiapkan aturan untuk mencocokkan peristiwa (seperti DeleteObject permintaan) dan merutekannya ke satu atau beberapa fungsi atau pengaliran target. CloudWatch Peristiwa menjadi sadar akan perubahan operasional yang terjadi. Selain itu, CloudWatch Events merespons perubahan operasional ini dan mengambil tindakan korektif seperlunya, dengan mengirim pesan untuk merespons lingkungan, mengaktifkan fungsi, membuat perubahan, dan menangkap informasi status.

## CloudWatch Log

Pencatatan akses menyediakan catatan terperinci untuk permintaan yang dilakukan ke objek dalam wadah. Log akses berguna untuk banyak aplikasi, seperti audit keamanan dan akses. Mereka juga dapat membantu Anda mempelajari basis pelanggan Anda dan memahami MediaStore tagihan Anda. CloudWatch Log dikategorikan sebagai berikut:

- Pengaliran log adalah urutan log acara yang berbagi sumber yang sama.
- Grup log adalah grup pengaliran log yang berbagi pengaturan retensi, pemantauan, dan kontrol akses yang sama. Saat Anda mengaktifkan akses masuk pada wadah, MediaStore buat grup log dengan nama seperti `/aws/mediastore/MyContainerName`. Anda dapat menentukan grup log dan menentukan pengaliran untuk dimasukkan ke dalam setiap grup. Tidak ada kuota pada jumlah pengaliran log yang dapat menjadi milik satu grup log.

Secara default, log disimpan tanpa batas waktu dan tidak pernah kedaluwarsa. Anda dapat menyesuaikan kebijakan retensi untuk setiap grup log, menjaga retensi yang tak terbatas, atau memilih periode retensi untuk setiap grup log, menjaga retensi yang tak terbatas, atau memilih periode retensi dari satu hari hingga 10 tahun.

## Mengatur izin untuk Amazon CloudWatch

Gunakan AWS Identity and Access Management (IAM) untuk membuat peran yang memberikan MediaStore akses AWS Elemental ke Amazon CloudWatch. Anda harus melakukan langkah-langkah ini agar CloudWatch Log dipublikasikan untuk akun Anda. CloudWatch secara otomatis menerbitkan metrik untuk akun Anda.

Untuk mengizinkan MediaStore akses ke CloudWatch

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi konsol IAM, pilih Kebijakan, lalu pilih Buat kebijakan.
3. Pilih tab JSON dan tempel kebijakan berikut ini:

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

Kebijakan ini memungkinkan MediaStore untuk membuat grup log dan aliran log untuk setiap kontainer di Wilayah mana pun dalam AWS akun Anda.

4. Pilih Tinjau kebijakan.

5. Pada halaman Kebijakan tinjauan, untuk Nama, masukkan **MediaStoreAccessLogsPolicy**, lalu pilih Buat kebijakan.
6. Di panel navigasi konsol IAM, pilih Peran, dan lalu pilih Buat peran.
7. Pilih jenis peran Akun AWS lainnya.
8. Untuk ID Akun, masukkan IDAWS akun Anda.
9. Pilih Next: Permissions (Selanjutnya: Izin).
10. Dalam kotak pencarian, masukkan **MediaStoreAccessLogsPolicy**.
11. Pilih kotak centang di samping kebijakan baru, lalu pilih Berikutnya: Tag.
12. Pilih Berikutnya: Tinjau untuk melihat pratinjau pengguna baru Anda.
13. Untuk Nama peran, masukkan **MediaStoreAccessLogs**, lalu pilih Buat peran.
14. Di pesan konfirmasi, pilih nama peran yang baru saja Anda buat (**MediaStoreAccessLogs**).
15. Pada halaman Ringkasan peran, pilih tab Hubungan kepercayaan.
16. Pilih Edit trust relationship (Edit Hubungan Kepercayaan).
17. Dalam dokumen kebijakan, ubah kepala sekolah ke MediaStore layanan. Seharusnya terlihat seperti ini:

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

Seluruh kebijakan harus dibaca sebagai berikut:

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

18. Pilih Perbarui Kebijakan Kepercayaan.

## Mengaktifkan pencatatan akses untuk sebuah kontainer

Secara default, AWS Elemental MediaStore tidak mengumpulkan log akses. Saat Anda mengaktifkan pencatatan akses pada kontainer, MediaStore mengirimkan log akses untuk objek yang disimpan di kontainer tersebut ke Amazon CloudWatch. Log akses menyediakan catatan terperinci untuk permintaan yang dilakukan ke setiap objek yang disimpan dalam wadah. Informasi ini dapat mencakup jenis permintaan, sumber daya yang ditentukan dalam permintaan, dan waktu serta tanggal pemrosesan permintaan.

### Important

Tidak ada biaya tambahan untuk mengaktifkan pencatatan akses di MediaStore kontainer. Namun, semua file log yang dikirim layanan kepada Anda mengumpulkan biaya penyimpanan biasa. (Anda dapat menghapus berkas log ini kapan saja.) AWS tidak menilai biaya transfer data untuk pengiriman berkas log, tetapi memang mengenakan tarif transfer data normal untuk mengakses berkas log.

Untuk mengaktifkan pengelogan akses (AWS CLI)

- Dalam AWS CLI, gunakan `start-access-logging` perintah:

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

Perintah ini tidak memiliki nilai pengembalian.

## Menonaktifkan pengelogan akses untuk kontainer

Saat Anda menonaktifkan pencatatan akses pada kontainer, AWS Elemental MediaStore berhenti mengirim log akses ke Amazon CloudWatch. Log akses ini tidak disimpan dan tidak dapat diambil.

Untuk menonaktifkan pengelogan akses (AWS CLI)

- Dalam AWS CLI, gunakan `stop-access-logging` perintah:

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

Perintah ini tidak memiliki nilai pengembalian.

## Pemecahan masalah akses logging di AWS Elemental MediaStore

Jika log MediaStore akses AWS Elemental tidak muncul di Amazon CloudWatch, lihat tabel berikut untuk potensi penyebab dan resolusi.

### Note

Pastikan untuk mengaktifkan AWS CloudTrail Log untuk membantu proses pemecahan masalah.

Gejala	Masalahnya Mungkin...	Coba ini...
Anda tidak melihat CloudTrail peristiwa apa pun, meskipun CloudTrail log diaktifkan.	Peran IAM tidak ada atau memiliki nama, izin, atau kebijakan kepercayaan yang salah.	Buat peran dengan nama, izin, dan kebijakan kepercayaan yang benar. Lihat <a href="#">the section called “Mengatur izin untuk CloudWatch”</a> .
Anda mengirimkan permintaan <code>DescribeContainer</code> API, tetapi respons menunjukkan bahwa <code>AccessLoggingEnabled</code> parameter memiliki nilai <code>False</code> . Selain itu, Anda tidak melihat CloudTrail acara apa pun untuk <code>MediaStoreAccessLogs</code> peran yang berhasil <code>DescribeLogGroup</code> , <code>CreateLogGroup</code> , <code>DescribeLogStream</code> , atau <code>CreateLogStream</code> panggilan.	Peran IAM tidak ada atau memiliki nama, izin, atau kebijakan kepercayaan yang salah.  Pencatatan akses tidak diaktifkan pada kontainer.	Buat peran dengan nama, izin, dan kebijakan kepercayaan yang benar. Lihat <a href="#">the section called “Mengatur izin untuk CloudWatch”</a> .  Aktifkan log akses untuk wadah. Lihat <a href="#">the section called “Mengaktifkan pengelogan akses”</a> .
Di CloudTrail konsol, Anda melihat peristiwa dengan kesalahan akses ditolak yang terkait dengan <code>MediaStoreAccessLogs</code> peran tersebut.	Peran IAM tidak memiliki izin yang benar untuk AWS Elemental MediaStore.	Perbarui peran IAM untuk memiliki izin yang benar dan kebijakan kepercayaan. Lihat <a href="#">the section called “Mengatur izin untuk CloudWatch”</a> .

Gejala	Masalahnya Mungkin...	Coba ini...
<p>CloudTrail Acara ini mungkin mencakup baris seperti berikut ini:</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>Anda tidak melihat log untuk seluruh kontainer atau kontainer.</p>	<p>Akun Anda mungkin telah melampaui CloudWatch kuota untuk grup log per Wilayah. Lihat kuota untuk grup log di <a href="#">Panduan Pengguna Amazon CloudWatch Logs</a>.</p>	<p>Di CloudWatch konsol, tentukan apakah akun Anda telah memenuhi CloudWatch kuota untuk grup log. Jika perlu, <a href="#">minta kenaikan kuota</a>.</p>

Gejala	Masalahnya Mungkin...	Coba ini...
Anda melihat beberapa log in CloudWatch, tetapi tidak semua log yang Anda harapkan untuk melihat.	Akun Anda mungkin telah melebihi CloudWatch kuota per Wilayah. Lihat kuota untuk PutLogEvents di <a href="#">Panduan Pengguna Amazon CloudWatch Logs</a> .	<a href="#">Meminta CloudWatch kenaikan kuota</a> per Wilayah.

## Format log akses

Berkas log akses terdiri atas urutan catatan log yang diformat JSON, di mana setiap catatan log mewakili satu permintaan. Urutan bidang dalam log dapat bervariasi. Berikut ini contoh log contoh yang terdiri dari dua catatan 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
}
```

Daftar berikut menjelaskan bidang catatan log:

#### AWSAccountId

ID AWS akun dari akun yang digunakan untuk membuat permintaan.

#### BytesReceived

Jumlah byte dalam badan permintaan yang diterima MediaStore server.

#### BytesSent

Jumlah byte dalam tubuh respon yang dikirim MediaStore server. Nilai ini sering sama dengan nilai `Content-Length` header yang disertakan dengan respons server.

#### ContainerName

Nama kontainer yang menerima permintaan tersebut.

#### ErrorCode

Kode MediaStore kesalahan (seperti `InternalServerError`). Jika tidak ada kesalahan terjadi, karakter muncul. Kode kesalahan mungkin muncul meskipun kode status 200 (menunjukkan koneksi tertutup atau kesalahan setelah server mulai mengalirkan respons).

#### ExpiresAt

Tanggal kedaluwarsa objek dan waktu. Nilai ini didasarkan pada usia [transient data rule](#) kedaluwarsa yang ditetapkan oleh kebijakan siklus hidup yang diterapkan ke kontainer. Nilainya



adalah waktu ISO-8601 tanggal dan didasarkan pada jam sistem host yang melayani permintaan. Jika kebijakan siklus hidup tidak memiliki aturan data sementara yang berlaku untuk objek, atau jika tidak ada kebijakan siklus hidup yang diterapkan ke kontainer, nilai bidang ini adalah null. Bidang ini hanya berlaku untuk operasi berikut: PutObject, GetObject, DescribeObject, dan DeleteObject.

### HttpStatus

Kode status HTTP numerik dari respons.

### Operasi

Operasi yang dilakukan, seperti PutObject atau ListItems.

### Jalur

Jalan dalam wadah di mana objek disimpan. Jika operasi tidak mengambil parameter path, - karakter muncul.

### ReceivedTime

Waktu hari ketika permintaan diterima. Nilainya adalah waktu ISO-8601 tanggal dan didasarkan pada jam sistem host yang melayani permintaan.

### Pemohon

Amazon Resource Name (ARN) dari akun yang digunakan untuk membuat permintaan. Untuk permintaan yang tidak diautentikasi, nilai ini adalah anonymous. Jika permintaan gagal sebelum otentikasi selesai, bidang ini mungkin hilang dari log. Untuk permintaan tersebut, ErrorCode mungkin mengidentifikasi masalah otorisasi.

### memintaId

String yang dibuat oleh AWS Elemental MediaStore untuk mengidentifikasi setiap permintaan secara unik.

### Sumber

Alamat internet yang jelas dari pemohon atau kepala layanan layanan AWS yang melakukan panggilan. Jika proxy perantara dan firewall mengaburkan alamat mesin yang membuat permintaan, nilai diatur ke null.

### TotalTime

Jumlah milidetik (ms) yang permintaan dalam proses pengiriman dari perspektif server. Nilai ini diukur dimulai dengan waktu permintaan Anda diterima oleh layanan dan berakhir dengan

waktu byte terakhir respons dikirim. Nilai ini diukur dari perspektif server karena pengukuran yang dilakukan dari perspektif klien dipengaruhi oleh latensi jaringan.

## TurnAroundTime

Jumlah milidetik yang MediaStore dihabiskan untuk memproses permintaan Anda. Nilai ini diukur dari waktu byte terakhir permintaan Anda diterima hingga saat byte pertama respons dikirim.

Urutan bidang dalam log dapat bervariasi.

## Perubahan status logging akan berlaku dari waktu

Perubahan status pencatatan log dari wadah memerlukan waktu untuk benar-benar memengaruhi pengiriman berkas log. Misalnya, jika Anda mengaktifkan pencatatan log untuk kontainer A, beberapa permintaan yang dilakukan di jam berikutnya mungkin dicatat, sementara yang lainnya mungkin tidak. Jika Anda menonaktifkan pencatatan untuk kontainer B, beberapa log untuk satu jam berikutnya mungkin terus dikirimkan, sementara yang lain mungkin tidak. Dalam semua kasus, pengaturan baru akhirnya berlaku tanpa tindakan lebih lanjut dari pihak Anda.

## Pengiriman log server dengan upaya terbaik

Catatan log akses disampaikan atas dasar upaya terbaik. Sebagian besar permintaan kontainer yang dikonfigurasi dengan benar untuk mencatat hasil dalam catatan log yang dikirim. Sebagian besar catatan log dikirim dalam beberapa jam setelah log dicatat, tetapi dapat dikirimkan lebih sering.

Kelengkapan dan ketepatan waktu pencatatan akses tidak dijamin. Catatan log untuk permintaan tertentu mungkin dikirim dalam waktu lama setelah permintaan diproses, atau mungkin tidak dikirimkan sama sekali. Tujuan log akses adalah untuk memberi Anda ide tentang sifat lalu lintas terhadap wadah Anda. Sangat jarang kehilangan catatan log, tetapi pencatatan akses tidak dimaksudkan untuk menjadi pencatat lengkap semua permintaan.

Proses ini mengikuti sifat upaya terbaik dari fitur pencatatan log akses yang laporan penggunaan yang tersedia di portal AWS (laporan Billing and Cost Management pada [AWS Management Console](#)) dapat mencakup satu atau beberapa permintaan akses yang tidak muncul di log akses yang dikirim.

## Pertimbangan pemrograman untuk format log akses

Dari waktu ke waktu, kami dapat memperluas format log akses dengan menambahkan bidang baru. Kode yang menguraikan log akses harus ditulis untuk menangani bidang tambahan yang tidak dipahaminya.

## CloudWatch Event

Amazon CloudWatch Events memungkinkan Anda mengotomatiskan AWS layanan Anda dan merespons peristiwa sistem secara otomatis seperti masalah ketersediaan aplikasi atau perubahan sumber daya. Anda dapat menulis aturan sederhana untuk menunjukkan kejadian mana yang sesuai kepentingan Anda, dan tindakan otomatis apa yang diambil ketika suatu kejadian sesuai dengan suatu aturan.

### Important

Biasanya, AWS layanan mengirimkan pemberitahuan peristiwa ke CloudWatch Acara dalam hitungan detik, tetapi terkadang perlu waktu satu menit atau lebih.

Saat file diunggah ke wadah atau dihapus dari kontainer, dua peristiwa dipecah secara berurutan dalam CloudWatch layanan:

1. [the section called “Peristiwa perubahan status objek”](#)
2. [the section called “Peristiwa perubahan status kontainer”](#)

Untuk informasi tentang berlangganan ke peristiwa ini, lihat [Amazon CloudWatch](#).

Tindakan yang dapat dipicu secara otomatis meliputi hal berikut:

- Mengambil fungsi AWS Lambda
- Meminta Perintah Amazon EC2 Run
- Mengirim peristiwa ke Amazon Kinesis Data Streams
- Mengaktifkan mesin keadaan AWS Step Functions
- Memberi tahu topik Amazon SNS atau AWS SMS antrean

Beberapa contoh penggunaan CloudWatch Events dengan AWS Elemental MediaStore adalah sebagai berikut:

- Mengaktifkan fungsi Lambda setiap kali kontainer dibuat
- Memberi tahu topik Amazon SNS saat objek dihapus

Untuk informasi selengkapnya, lihat [Panduan Pengguna Amazon CloudWatch Events](#).

## Topik

- [Peristiwa perubahan status MediaStore objek AWS Elemental](#)
- [Peristiwa perubahan status MediaStore kontainer AWS Elemental](#)

## Peristiwa perubahan status MediaStore objek AWS Elemental

Acara ini diterbitkan ketika negara objek telah berubah (ketika objek telah diunggah atau dihapus).

### Note

Objek yang kedaluwarsa karena aturan data sementara tidak memancarkan CloudWatch peristiwa saat kedaluwarsa.

Untuk informasi tentang berlangganan ke peristiwa ini, lihat [Amazon CloudWatch](#).

## Objek diperbarui

```
{
  "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"
  }
}
```

## Objek dihapus

```
{
  "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"
  }
}
```

## Peristiwa perubahan status MediaStore kontainer AWS Elemental

Peristiwa ini diterbitkan ketika status kontainer telah berubah (ketika wadah telah ditambahkan atau dihapus). Untuk informasi tentang berlangganan ke peristiwa ini, lihat [Amazon CloudWatch](#).

## Wadah dibuat

```
{
  "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"
}
}
```

## Wadah dihapus

```
{
  "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"
  }
}
```

## Memantau AWS Elemental MediaStore dengan CloudWatch metrik Amazon

Anda dapat memantau MediaStore penggunaan AWS Elemental CloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik yang dapat dibaca. CloudWatch membuat statistik disimpan untuk jangka waktu 15 bulan, sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang performa aplikasi web atau layanan Anda. Anda juga dapat mengatur alarm yang memperhatikan ambang batas tertentu dan mengirim notifikasi atau mengambil tindakan saat ambang batas tersebut terpenuhi. Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#).

Untuk AWS Elemental MediaStore, Anda mungkin ingin menonton `BytesDownloaded` dan mengirim email kepada diri Anda sendiri ketika metrik tersebut mencapai ambang batas tertentu.

Untuk melihat metrik menggunakan CloudWatch konsol

Metrik dikelompokkan terlebih dahulu berdasarkan namespace layanan, kemudian berdasarkan berbagai kombinasi dimensi dalam setiap namespace.

1. Masuk ke AWS Management Console dan buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Di panel navigasi, pilih Metrik.
3. Di bawah Semua metrik, pilih MediaStoreAWS/namespaces.
4. Pilih dimensi metrik untuk melihat metrik. Misalnya, pilih `Request metrics by container` untuk melihat metrik untuk berbagai jenis permintaan yang telah dikirim ke wadah.

Untuk melihat metrik menggunakan AWS CLI

- Pada prompt perintah, gunakan perintah berikut:

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

## MediaStore Metrik AWS Elemental

Tabel berikut mencantumkan metrik yang MediaStore dikirim AWS Elemental CloudWatch.

### Note

Untuk melihat metrik, Anda harus [menambahkan kebijakan metrik](#) ke wadah MediaStore untuk memungkinkan mengirim metrik ke Amazon CloudWatch.

Metrik	Deskripsi
RequestCount	<p>Jumlah total permintaan HTTP yang dibuat untuk MediaStore wadah, dipisahkan oleh jenis operasi (Put, Get, Delete, Describe, List).</p> <p>Unit: Count (Jumlah)</p> <p>Dimensi yang valid:</p> <ul style="list-style-type: none"> <li>• Nama kontainer</li> <li>• Nama grup objek</li> <li>• Jenis permintaan</li> </ul>

Metrik	Deskripsi
	Statistik yang valid: Sum
4xxErrorCount	<p>Jumlah permintaan HTTP yang dibuat untuk MediaStore yang mengakibatkan kesalahan 4xx.</p> <p>Unit: Count (Jumlah)</p> <p>Dimensi yang valid:</p> <ul style="list-style-type: none"><li>• Nama kontainer</li><li>• Nama grup objek</li><li>• Jenis permintaan</li></ul> <p>Statistik yang valid: Sum</p>
5xxErrorCount	<p>Jumlah permintaan HTTP yang dibuat untuk MediaStore yang mengakibatkan kesalahan 5xx.</p> <p>Unit: Count (Jumlah)</p> <p>Dimensi yang valid:</p> <ul style="list-style-type: none"><li>• Nama kontainer</li><li>• Nama grup objek</li><li>• Jenis permintaan</li></ul> <p>Statistik yang valid: Sum</p>



Metrik	Deskripsi
BytesUploaded	<p>Jumlah byte yang diunggah untuk permintaan yang dibuat ke MediaStore wadah, di mana permintaan mencakup satu bodi.</p> <p>Unit: Bit</p> <p>Dimensi yang valid:</p> <ul style="list-style-type: none"><li>• Nama kontainer</li><li>• Nama grup objek</li></ul> <p>Rata-rata statistik yang valid (byte per permintaan), Jumlah (byte per periode), Hitungan Sampel, Min (sama dengan P0.0), Maks (sama dengan p100), setiap persentil antara p99.9</p>
BytesDownloaded	<p>Jumlah byte yang diunduh untuk permintaan yang dibuat ke MediaStore wadah, yang tanggapannya mencakup satu bodi.</p> <p>Unit: Bit</p> <p>Dimensi yang valid:</p> <ul style="list-style-type: none"><li>• Nama kontainer</li><li>• Nama grup objek</li></ul> <p>Rata-rata statistik yang valid (byte per permintaan), Jumlah (byte per periode), Hitungan Sampel, Min (sama dengan P0.0), Maks (sama dengan p100), setiap persentil antara p99.9</p>

Metrik	Deskripsi
TotalTime	<p>Jumlah milidetik permintaan dalam proses pengiriman dari perspektif server. Nilai ini diukur dari waktu yang MediaStore menerima permintaan Anda, hingga waktu yang dikirim byte terakhir respons. Nilai ini diukur dari perspektif server karena pengukuran yang dilakukan dari perspektif klien dipengaruhi oleh latensi jaringan.</p> <p>Unit: Milidetik</p> <p>Dimensi yang valid:</p> <ul style="list-style-type: none"><li>• Nama kontainer</li><li>• Nama grup objek</li><li>• Jenis permintaan</li></ul> <p>Rata-rata statistik yang valid, Min (sama dengan P0,0), Maks (sama dengan p100), setiap persentil antara p0,0 dan p100</p>
TurnaroundTime	<p>Jumlah milidetik yang MediaStore dihabiskan untuk memproses permintaan Anda. Nilai ini diukur dari waktu yang MediaStore menerima byte terakhir permintaan Anda, hingga saat mengirim byte pertama respons.</p> <p>Unit: Milidetik</p> <p>Dimensi yang valid:</p> <ul style="list-style-type: none"><li>• Nama kontainer</li><li>• Nama grup objek</li><li>• Jenis permintaan</li></ul> <p>Rata-rata statistik yang valid, Min (sama dengan P0,0), Maks (sama dengan p100), setiap persentil antara p0,0 dan p100</p>

Metrik	Deskripsi
ThrottledCount	<p>Jumlah permintaan HTTP yang dibuat untuk MediaStore yang dicekatkan.</p> <p>Unit: Count (Jumlah)</p> <p>Dimensi yang valid:</p> <ul style="list-style-type: none"> <li>• Nama kontainer</li> <li>• Nama grup objek</li> <li>• Jenis permintaan</li> </ul> <p>Statistik yang valid: Sum</p>

## Menandai sumber daya MediaStore AWS Elemental

Tag merupakan label atribut kustom yang Anda atau AWS tetapkan ke sumber daya AWS. Setiap tanda memiliki dua bagian:

- Sebuah kunci tag (misalnya, `CostCenter`, `Environment`, atau `Project`). Kunci tag peka terhadap huruf besar dan kecil.
- Bidang opsional yang dikenal sebagai nilai tanda (misalnya, `111122223333` atau `Production`). Mengabaikan nilai tag sama saja dengan menggunakan string kosong. Seperti kunci tag, nilai tag peka huruf besar dan kecil.

Tag membantu Anda melakukan hal berikut:

- Identifikasi dan organisir sumber daya AWS Anda. Banyak tag memberikan support pada layanan AWS, sehingga Anda dapat menetapkan tag yang sama ke sumber daya dari berbagai layanan untuk menunjukkan bahwa sumber daya tersebut terkait. Misalnya, Anda dapat menetapkan tag yang sama ke wadah AWS MediaStore *Elemental* yang Anda tetapkan ke input. AWS Elemental MediaLive
- Telusuri biaya AWS Anda. Anda mengaktifkan tag ini di AWS Billing and Cost Management dasbor. AWS menggunakan tag untuk mengategorikan biaya Anda dan mengirimkan laporan alokasi biaya bulanan kepada Anda. Untuk informasi selengkapnya, lihat [Gunakan Tag Alokasi Biaya](#) dalam [AWS BillingPanduan Pengguna](#).

Bagian berikut memberikan informasi selengkapnya tentang tag untuk AWS Elemental MediaStore.

## Sumber daya yang didukung di AWS Elemental MediaStore

Sumber daya berikut dalam AWS Elemental MediaStore mendukung penandaan:

- *kontainer*

Untuk informasi tentang menambahkan dan mengelola tag, lihat [Mengelola tag](#).

AWS Elemental MediaStore tidak mendukung fitur kontrol akses berbasis tag AWS Identity and Access Management (IAM).

## Konvensi penamaan dan penggunaan tag

Konvensi penamaan dan penggunaan dasar berikut berlaku untuk menggunakan tag dengan sumber daya AWS MediaStore Elemental:

- Setiap sumber daya dapat memiliki maksimum tanda 50.
- Untuk setiap sumber daya, setiap tanda kunci harus unik, dan setiap tanda kunci hanya dapat memiliki satu nilai.
- Panjang tanda kunci maksimum adalah 128 karakter Unicode dalam UTF-8.
- Panjang tanda nilai maksimum adalah 256 karakter Unicode dalam UTF-8.
- Karakter yang diperbolehkan adalah huruf, angka, spasi yang dapat ditampilkan di UTF-8, serta karakter berikut: . : + = @ \_ / - (tanda hubung). Sumber daya Amazon EC2 memungkinkan karakter apa pun.
- Kunci dan nilai tag peka huruf besar dan kecil. Sebagai praktik terbaik, putuskan strategi untuk memanfaatkan tag dan terapkan strategi tersebut secara konsisten di semua jenis sumber daya. Misalnya, putuskan apakah akan menggunakan Costcenter, costcenter, atau CostCenter dan menggunakan kesepakatan yang sama untuk semua tag. Hindari penggunaan tag yang serupa dengan perlakuan kasus yang tidak konsisten.
- `aws :` Awalan dilarang untuk tag; itu dicadangkan untuk AWS digunakan. Anda tidak dapat mengedit atau menghapus kunci atau nilai tanda dengan prefiks ini. Tanda dengan prefiks ini tidak memengaruhi tanda Anda per kuota sumber daya.

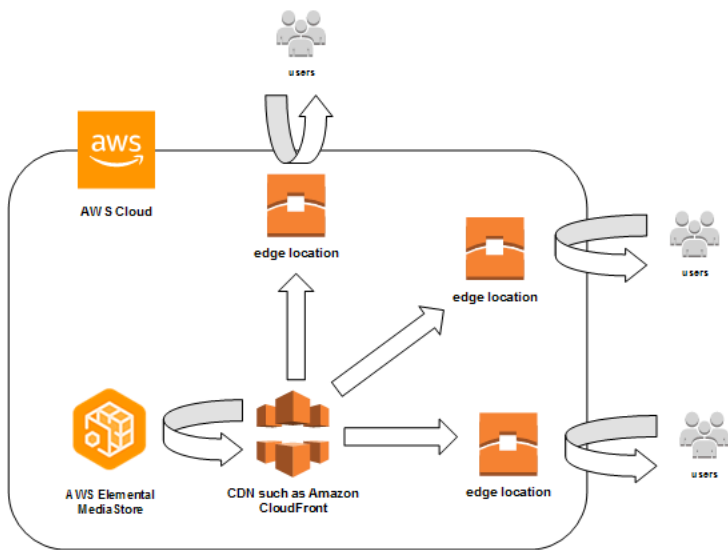
## Mengelola tag

Tag terdiri dari Key dan Value properti pada sumber daya. Anda dapat menggunakan AWS CLI atau MediaStore API untuk menambahkan, mengedit, atau menghapus nilai untuk properti ini. Untuk informasi tentang bekerja dengan tag, lihat bagian berikut di Referensi AWS Elemental MediaStore API:

- [CreateContainer](#)
- [ListTagsForResource](#)
- [Sumber Daya](#)
- [TagResource](#)
- [UntagResource](#)

## Bekerja dengan jaringan pengiriman konten (CDN)

Anda dapat menggunakan jaringan pengiriman konten (CDN) seperti [Amazon CloudFront](#) untuk menyajikan konten yang Anda simpan di AWS Elemental MediaStore. CDN adalah kumpulan server yang didistribusikan secara global yang menyimpan konten seperti video. Saat pengguna meminta konten Anda, CDN merutekan permintaan ke lokasi edge yang menyediakan latensi terendah. Jika konten Anda sudah di-cache di lokasi edge dengan, CDN segera mengirimkannya. Jika konten Anda saat ini tidak berada di lokasi edge tersebut, CDN mengambilnya dari asal Anda (seperti MediaStore wadah Anda) dan mendistribusikannya ke pengguna.



### Topik

- [Mengizinkan Amazon CloudFront mengakses MediaStore kontainer AWS Elemental](#)
- [Interaksi AWS Elemental MediaStore dengan cache HTTP](#)

## Mengizinkan Amazon CloudFront mengakses MediaStore kontainer AWS Elemental

Anda dapat menggunakan Amazon CloudFront untuk menyajikan konten yang Anda simpan dalam wadah di AWS Elemental MediaStore. Anda dapat melakukannya melalui salah satu cara berikut:

- [Menggunakan Origin Access Control \(OAC\)](#)- (Direkomendasikan) Gunakan opsi ini jika AndaWilayah AWS mendukung fitur OAC CloudFront.

- [Menggunakan rahasia bersama](#)- Gunakan opsi ini jika AndaWilayah AWS tidak mendukung fitur OAC CloudFront.

## Menggunakan Origin Access Control (OAC)

Anda dapat menggunakan fitur Origin Access Control (OAC) Amazon CloudFront untuk mengamankan MediaStore asal AWS Elemental dengan keamanan yang ditingkatkan. Anda dapat mengaktifkan [AWSSignature Version 4 \(SiGv4\)](#) pada CloudFront permintaan MediaStore asal dan mengatur kapan dan jika CloudFront harus menandatangani permintaan. Anda dapat mengakses fitur OAC CloudFront melalui konsol, API, SDK, atau CLI, dan tidak ada biaya tambahan untuk penggunaannya.

Untuk informasi selengkapnya tentang menggunakan fitur OAC MediaStore, lihat [Membatasi akses ke MediaStore asal](#) di [Panduan CloudFront Pengembang Amazon](#).

## Menggunakan rahasia bersama

Jika AndaWilayah AWS tidak mendukung fitur OAC Amazon CloudFront, Anda dapat melampirkan kebijakan ke MediaStore wadah AWS Elemental yang memberikan akses baca atau lebih besar CloudFront.

### Note

Sebaiknya gunakan fitur OAC jika AndaWilayah AWS mendukungnya. Prosedur berikut mengharuskan Anda untuk mengkonfigurasi MediaStore dan CloudFront dengan rahasia bersama untuk membatasi akses ke MediaStore kontainer. Untuk mengikuti praktik keamanan terbaik, konfigurasi manual ini memerlukan rotasi rahasia secara berkala. Dengan OAC on MediaStore origins, Anda dapat menginstruksikan CloudFront untuk menandatangani permintaan menggunakan SiGv4 dan meneruskannya MediaStore untuk pencocokan tanda tangan, menghilangkan kebutuhan untuk menggunakan dan memutar rahasia. Hal ini memastikan bahwa permintaan secara otomatis diverifikasi sebelum konten media disajikan, membuat pengiriman konten media melalui MediaStore dan CloudFront lebih sederhana dan lebih aman.

Untuk memungkinkan CloudFront untuk mengakses kontainer Anda (konsol)

1. Buka MediaStore konsol di <https://console.aws.amazon.com/mediastore/>.

2. Pada halaman Container, pilih nama kontainer.

Halaman detail kontainer akan muncul.

3. Di bagian Kebijakan Kontainer, lampirkan kebijakan yang memberikan akses baca atau lebih besar ke Amazon CloudFront.


### Example

Contoh kebijakan berikut, yang mirip dengan kebijakan contoh untuk [Akses Baca Publik melalui HTTPS](#), cocok dengan persyaratan ini karena memungkinkan `GetObject` dan `DescribeObject` perintah dari siapa saja yang mengirimkan permintaan ke domain Anda melalui HTTPS. Selanjutnya, kebijakan contoh berikut lebih baik mengamankan alur kerja Anda karena memungkinkan CloudFront akses ke MediaStore objek hanya ketika permintaan terjadi melalui koneksi HTTPS dan berisi header Referer yang benar.

```
{
  "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. Di bagian kebijakan Container CORS, tetapkan kebijakan yang memungkinkan tingkat akses yang sesuai.



 Note

[Kebijakan CORS](#) diperlukan hanya jika Anda ingin memberikan akses ke pemutar berbasis browser.

5. Catat detail berikut:
  - Titik akhir data yang ditetapkan untuk kontainer Anda. Anda dapat menemukan informasi ini di bagian Info pada halaman Containers. Di CloudFront, titik akhir data disebut sebagai nama domain asal.
  - Struktur folder dalam wadah tempat objek disimpan. Di CloudFront, ini disebut sebagai jalur asal. Perhatikan bahwa pengaturan ini bersifat opsional. Untuk informasi selengkapnya tentang jalur asal, lihat [Panduan CloudFront Pengembang Amazon](#).
6. Di dalamnya CloudFront, buat distribusi yang [dikonfigurasi untuk menyajikan konten dari AWS Elemental MediaStore](#). Anda akan memerlukan informasi yang Anda kumpulkan pada langkah sebelumnya.

Setelah melampirkan kebijakan ke MediaStore kontainer Anda, Anda harus mengkonfigurasi CloudFront untuk hanya menggunakan koneksi HTTPS untuk permintaan asal, dan juga menambahkan header kustom dengan nilai rahasia yang benar.

Untuk mengkonfigurasi CloudFront untuk mengakses kontainer Anda melalui koneksi HTTPS dengan nilai rahasia untuk header Referer (konsol)

1. Buka CloudFront konsol.
2. Pada halaman Origins, pilih MediaStore asal Anda.
3. Pilih Edit.
4. Pilih HTTPS hanya untuk protokol.
5. Dalam Tambahkan header kustom bagian, pilih Tambahkan header.
6. Untuk Nama, pilih Referer. Untuk nilainya, gunakan `<secretValue>string` yang sama dengan yang Anda gunakan dalam kebijakan kontainer.
7. Pilih Simpan dan biarkan perubahan diterapkan.

## Interaksi AWS Elemental MediaStore dengan cache HTTP

AWS Elemental MediaStore menyimpan objek sehingga objek dapat di-cache dengan benar dan efisien oleh jaringan pengiriman konten (CDN) seperti Amazon CloudFront. Ketika pengguna akhir atau CDN mengambil objek dari MediaStore, layanan mengembalikan header HTTP yang mempengaruhi perilaku caching objek. (Standar untuk HTTP 1.1 perilaku caching ditemukan di [RFC2616 bagian 13.](#)) Header ini adalah:

- **ETag**(tidak dapat disesuaikan) - Header tag entitas adalah pengenalan unik untuk respons yang MediaStore dikirim. CDN dan browser web yang sesuai standar menggunakan tag ini sebagai kunci untuk menyimpan objek. MediaStore secara otomatis menghasilkan ETag untuk setiap objek ketika di-upload. Anda dapat [melihat detail objek](#) untuk menentukan nilai ETag.
- **Last-Modified**(tidak dapat disesuaikan) - Nilai header ini menunjukkan tanggal dan waktu objek dimodifikasi. MediaStore secara otomatis menghasilkan nilai ini ketika objek diunggah.
- **Cache-Control**(dapat disesuaikan) - Nilai header ini mengontrol berapa lama objek harus di-cache sebelum CDN memeriksa untuk melihat apakah telah dimodifikasi. Anda dapat mengatur header ini ke nilai apa pun saat Anda mengunggah objek ke MediaStore wadah menggunakan [CLI](#) atau [API](#). Kumpulan lengkap nilai valid dijelaskan dalam [dokumentasi HTTP/1.1](#). Jika Anda tidak menetapkan nilai ini ketika Anda meng-upload objek, MediaStore tidak akan kembali header ini ketika objek diambil.

Kasus penggunaan umum untuk header Cache-Control adalah menentukan durasi untuk cache objek. Misalnya, Anda memiliki file manifes video yang sedang sering ditimpa oleh encoder. Anda dapat mengatur `max-age` ke 10 untuk menunjukkan bahwa objek harus di-cache hanya selama 10 detik. Atau anggaplah Anda memiliki segmen video tersimpan yang tidak akan pernah ditimpa. Anda dapat mengatur `max-age` untuk objek ini ke 31536000 ke cache selama kurang lebih 1 tahun.

## Permintaan bersyarat

### Permintaan bersyarat untuk MediaStore

MediaStore merespons identik permintaan bersyarat (menggunakan header permintaan seperti `If-Modified-Since` dan `If-None-Match`, seperti yang dijelaskan dalam [RFC7232](#)) dan permintaan tanpa syarat. Ini berarti bahwa ketika MediaStore menerima `GetObject` permintaan yang valid, layanan selalu mengembalikan objek bahkan jika klien sudah memiliki objek.

## Permintaan bersyarat ke CDN

CDN yang melayani konten atas nama MediaStore dapat memproses permintaan bersyarat dengan mengembalikan `304 Not Modified`, seperti yang dijelaskan dalam [RFC7232 bagian 4.1](#). Hal ini menunjukkan bahwa tidak perlu mentransfer isi objek lengkap, karena pemohon sudah memiliki objek yang cocok dengan permintaan kondisional.

CDN (dan cache lain yang sesuai dengan HTTP/1.1) mendasarkan keputusan ini pada `ETag` dan `Cache-Control` header yang diteruskan oleh server asal. Untuk mengontrol seberapa sering CDN meminta server MediaStore asal untuk pembaruan ke objek yang diambil berulang kali, atur `Cache-Control` header untuk objek tersebut saat Anda mengunggahnya MediaStore.

## Kuota di AWS Elemental MediaStore

Konsol Service Quotas menyediakan informasi tentang MediaStore kuota AWS Elemental. Sembari melihat kuota default, Anda dapat menggunakan konsol Service Quotas guna [mengajukan penambahan kuota](#) untuk kuota yang dapat disesuaikan.

Tabel berikut menjelaskan kuota, sebelumnya disebut sebagai batas, di AWS Elemental MediaStore. Kuota adalah jumlah maksimum sumber daya layanan atau operasi untuk akun AWS Anda.

### Note

Untuk menetapkan kuota ke kontainer individual dalam akun Anda, hubungi AWS Support atau manajer akun Anda. Opsi ini dapat membantu Anda membagi batas tingkat akun di antara kontainer Anda, untuk mencegah satu kontainer menggunakan seluruh kuota Anda.

Sumber Daya atau Operasi	Kuota Default	Comments
Kontainer	100	Jumlah maksimum kontainer yang dapat Anda buat di akun ini.
Tingkat Folder	10	Jumlah maksimum tingkat folder yang dapat Anda buat dalam wadah. Anda dapat membuat folder sebanyak yang Anda inginkan, selama mereka tidak bersarang lebih dari 10 tingkat dalam wadah.
Folder	Tidak terbatas.	Anda dapat membuat folder sebanyak yang Anda inginkan, selama mereka tidak bersarang lebih dari 10 tingkat dalam wadah.
Ukuran Objek	25 MB	Ukuran file maksimum objek tunggal.
Objek	Tidak terbatas.	Anda dapat mengunggah objek sebanyak yang Anda inginkan ke folder atau wadah di akun Anda.

Sumber Daya atau Operasi	Kuota Default	Comments
Tingkat Permintaan API <a href="#">DeleteObject</a>	100	Jumlah maksimum permintaan operasi yang dapat Anda buat per detik. Permintaan tambahan diperlambat.  Anda dapat <a href="#">meminta penambahan kuota</a> .
Tingkat Permintaan API <a href="#">DescribeObject</a>	1.000	Jumlah maksimum permintaan operasi yang dapat Anda buat per detik. Permintaan tambahan diperlambat.  Anda dapat <a href="#">meminta penambahan kuota</a> .
Tingkat permintaan <a href="#">GetObject</a> API untuk ketersediaan pengunggahan standar	1.000	Jumlah maksimum permintaan operasi yang dapat Anda buat per detik. Permintaan tambahan diperlambat.  Anda dapat <a href="#">meminta penambahan kuota</a> .
Tingkat permintaan <a href="#">GetObject</a> API untuk ketersediaan pengunggahan streaming	25	Jumlah maksimum permintaan operasi yang dapat Anda buat per detik. Permintaan tambahan diperlambat.  Anda dapat <a href="#">meminta penambahan kuota</a> .
Tingkat Permintaan API <a href="#">ListItems</a>	5	Jumlah maksimum permintaan operasi yang dapat Anda buat per detik. Permintaan tambahan diperlambat.  Anda dapat <a href="#">meminta penambahan kuota</a> .

Sumber Daya atau Operasi	Kuota Default	Comments
Tingkat permintaan <a href="#">PutObjectAPI</a> untuk pengkodean transfer chunked (juga dikenal sebagai ketersediaan pengunggahan streaming)	10	Jumlah maksimum permintaan operasi yang dapat Anda buat per detik. Permintaan tambahan diperlambat.  Anda dapat <a href="#">meminta penambahan kuota</a> . Dalam permintaan, tentukan TPS yang diminta dan ukuran objek rata-rata.
Tingkat permintaan <a href="#">PutObjectAPI</a> untuk ketersediaan pengunggahan standar	100	Jumlah maksimum permintaan operasi yang dapat Anda buat per detik. Permintaan tambahan diperlambat.  Anda dapat <a href="#">meminta penambahan kuota</a> . Dalam permintaan, tentukan TPS yang diminta dan ukuran objek rata-rata.
Aturan dalam Kebijakan Metrik	10	Jumlah maksimum aturan yang dapat Anda sertakan dalam kebijakan metrik.
Aturan dalam Kebijakan Siklus Hidup Objek	10	Jumlah maksimum aturan yang dapat Anda sertakan dalam kebijakan siklus hidup objek.

# Informasi MediaStore terkait AWS Elemental

Berikut ini adalah tabel yang mencantumkan daftar sumber daya terkait yang akan berguna saat Anda bekerja dengan AWS Elemental MediaStore.

- [Kelas & Lokakarya](#) — Tautan ke kursus specialty dan berbasis peran, selain lab mandiri untuk membantu mempertajam AWS keterampilan Anda dan mendapatkan pengalaman praktis.
- [AWS Pusat Pengembang](#) - Jelajahi tutorial, alat unduh, dan pelajari tentang acara AWS pengembang.
- [AWS Alat Pengembang](#) — Tautan ke alat pengembang, SDK, kit alat IDE, dan alat baris perintah untuk mengembangkan dan mengelola AWS aplikasi.
- [Memulai Pusat Sumber Daya](#) — Pelajari cara mengatur Akun AWS, bergabung dengan AWS komunitas, dan meluncurkan aplikasi pertama Anda.
- [Tutorial Hands-On](#) - Ikuti step-by-step tutorial untuk meluncurkan aplikasi pertama Anda AWS.
- [AWS Laporan Resmi](#) — Tautan ke daftar lengkap AWS laporan resmi teknis, yang mencakup topik seperti arsitektur, keamanan, dan ekonomi dan ditulis oleh Arsitek AWS Solusi atau ahli teknis lainnya.
- [AWS Support Center](#) — Hub untuk membuat dan mengelola AWS Support kasus Anda. Juga mencakup tautan ke sumber daya yang bermanfaat lainnya, seperti forum, FAQ teknis, status kondisi layanan, dan AWS Trusted Advisor.
- [AWS Support](#) — Halaman web utama untuk informasi tentang AWS Support, saluran dukungan respons cepat untuk membantu Anda membangun dan menjalankan aplikasi di cloud. one-on-one
- [Kontak Kami](#) – Titik kontak pusat untuk pertanyaan tentang tagihan AWS, akun, peristiwa, penyalahgunaan, dan masalah lainnya.
- [AWS Persyaratan Situs](#) – Informasi detail tentang hak cipta dan merek dagang kami; akun, lisensi, dan akses situs Anda; serta topik lainnya.

## Riwayat dokumen untuk panduan pengguna

Tabel berikut menjelaskan dokumentasi untuk rilis AWS Elemental ini MediaStore. Untuk notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke umpan RSS.

Perubahan	Deskripsi	Tanggal
<a href="#">Peningkatan Kontrol Akses Asal (OAC)</a>	Menambahkan informasi tentang cara menggunakan OAC dengan AWS Elemental MediaStore.	17 April
<a href="#">Pembaruan kuota</a>	Dikoreksi nilai kuota dan deskripsi untuk <code>Rules in a Metric Policy</code> .	25 Oktober
<a href="#">ExpiresAt bidang</a>	Log akses sekarang menyertakan <code>ExpiresAt</code> bidang yang menunjukkan tanggal dan waktu kedaluwarsa objek berdasarkan aturan data sementara dalam kebijakan siklus hidup kontainer.	16 Juli
<a href="#">Aturan transisi siklus hidup</a>	Sekarang Anda dapat menambahkan aturan transisi siklus hidup ke kebijakan siklus hidup objek yang menetapkan objek untuk dipindahkan ke kelas penyimpanan akses jarang (IA) setelah mencapai usia tertentu.	20 April 2020



---

<a href="#">Wadah kosong</a>	Anda sekarang dapat menghapus semua objek dalam wadah sekaligus.	7 April 2020
<a href="#">Support untuk CloudWatch metrik Amazon</a>	Anda dapat menetapkan kebijakan metrik untuk menentukan metrik mana yang MediaStore dikirim CloudWatch.	30 Maret 2020
<a href="#">Wildcard dalam aturan objek hapus</a>	Dalam kebijakan siklus hidup objek, Anda sekarang dapat menggunakan wildcard dalam aturan objek delete. Hal ini memungkinkan Anda untuk menentukan file berdasarkan nama file atau ekstensi yang Anda ingin layanan untuk menghapus setelah beberapa hari.	20 Desember 2019
<a href="#">Kebijakan siklus hidup objek</a>	Sekarang Anda dapat menambahkan aturan ke kebijakan siklus hidup objek yang menunjukkan kedaluwarsa berdasarkan usia dalam hitungan detik.	13 September

[AWS CloudFormation untuk membuat wadah secara otomatis](#)

Anda sekarang dapat menggunakan AWS CloudFormation template untuk membuat wadah secara otomatis. AWS CloudFormation Templat mengelola data untuk lima tindakan API: membuat kontainer, mengatur pencatatan akses, memperbarui kebijakan kontainer default, menambahkan kebijakan siklus hidup objek.

17 Mei 2019

[Kuota untuk ketersediaan pengunggahan streaming](#)

Untuk objek dengan ketersediaan unggahan streaming (transfer objek yang dipotong), `PutObject` operasi tidak boleh melebihi 10 TPS dan `GetObject` operasi tidak boleh melebihi 25 TPS.

8 April 2019

[Pemindahan benda yang dipotong](#)

Ditambahkan dukungan untuk transfer chunked objek. Kemampuan ini memungkinkan Anda untuk menentukan bahwa objek tersedia untuk diunduh sebelum objek diunggah sepenuhnya.

5 April 2019

[Akses logging](#)

AWS Elemental MediaStore sekarang mendukung pencatatan akses, yang menyediakan catatan terperinci untuk permintaan yang dilakukan ke objek dalam kontainer.


25 Februari 2019

<a href="#">Kebijakan siklus hidup objek</a>	Menambahkan dukungan untuk kebijakan siklus hidup objek, yang mengatur tanggal kedaluwarsa objek dalam wadah saat ini.	12 Desember 2018
<a href="#">Peningkatan kuota ukuran objek</a>	Kuota untuk ukuran objek sekarang 25 MB.	10 Oktober 2018
<a href="#">Peningkatan kuota ukuran objek</a>	Kuota untuk ukuran objek sekarang 20 MB.	6 September 2018
<a href="#">AWS CloudTrail integrasi</a>	Konten CloudTrail integrasi telah diperbarui agar selaras dengan perubahan terbaru pada CloudTrail layanan.	12 Juli 2018
<a href="#">Kolaborasi CDN</a>	Menambahkan informasi tentang cara menggunakan AWS Elemental MediaStore dengan jaringan pengiriman konten (CDN) seperti Amazon CloudFront.	14 April
<a href="#">Konfigurasi CORS</a>	AWS Elemental MediaStore sekarang mendukung Cross-origin resource sharing (CORS), yang memungkinkan aplikasi web klien yang dimuat dalam satu domain untuk berinteraksi dengan sumber daya di domain yang berbeda.	Selasa, 07 Februari 2018

[Layanan dan panduan baru](#)

Ini adalah rilis awal layanan originasi dan penyimpanan video, AWS Elemental MediaStore, dan Panduan MediaStore Pengguna AWS Elemental.

27 November 2017

 Note

- Layanan AWS Media tidak dirancang atau dimaksudkan untuk digunakan dengan aplikasi atau dalam situasi yang membutuhkan kinerja yang gagal aman, seperti operasi keselamatan jiwa, sistem navigasi atau komunikasi, kontrol lalu lintas udara, atau mesin pendukung kehidupan di mana tidak tersedianya, gangguan, atau kegagalan layanan dapat menyebabkan kematian, cedera pribadi, kerusakan properti, atau kerusakan lingkungan.

# AWSGlosarium

Untuk AWS terminologi terbaru, lihat [AWSglosarium di Referensi](#). Glosarium AWS

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.