aws

Panduan Pengguna

AWSElemental MediaStore

AWSElemental MediaStore: Panduan Pengguna

Table of Contents

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

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 CloudFormationadalah 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 CloudTrailadalah layanan yang memungkinkan Anda memantau panggilan yang dilakukan ke CloudTrail API untuk akun Anda, termasuk panggilan yang dilakukan oleh AWS Management ConsoleAWS 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 jalankanAWS. 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 untuk mengakses AWS Elemental MediaStore. Untuk latar belakang dan informasi tambahan tentang identitas dan manajemen akses MediaStore, lihatIdentity and Access Management untuk AWS Elemental MediaStore.

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

Topik

- Mendaftar untuk Akun AWS
- Buat pengguna dengan akses administratif

Mendaftar untuk Akun AWS

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

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/pendaftaran.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan tindakan menerima panggilan telepon dan memasukkan kode verifikasi di keypad telepon.

   Ketika Anda mendaftar untuk Akun AWS, sebuah Pengguna root akun AWSdiciptakan. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

AWS mengirimi Anda email konfirmasi setelah proses pendaftaran selesai. Kapan saja, Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan masuk https://aws.amazon.comke/ dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftar untuk 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.

Amankan Pengguna root akun AWS

1. Masuk ke AWS Management Consolesebagai pemilik akun dengan memilih pengguna Root dan memasukkan Akun AWS alamat email. Di laman berikutnya, masukkan kata sandi.

   Untuk bantuan masuk menggunakan pengguna root, lihat Masuk sebagai pengguna root di AWS Sign-In Panduan Pengguna.
2. Aktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

   Untuk petunjuknya, lihat Mengaktifkan MFA perangkat virtual untuk Akun AWS root user (konsol) di Panduan IAM Pengguna.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat IAM Identitas.

   Untuk petunjuk, lihat Mengaktifkan AWS IAM Identity Centerdi AWS IAM Identity Center Panduan Pengguna.
2. Di Pusat IAM Identitas, berikan akses administratif ke pengguna.

   Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAMdi AWS IAM Identity Center Panduan Pengguna.

Masuk sebagai pengguna dengan akses administratif

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

  Untuk bantuan masuk menggunakan pengguna Pusat IAM Identitas, lihat Masuk ke AWS akses portal di AWS Sign-In Panduan Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat IAM Identitas, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

   Untuk petunjuknya, lihat Membuat set izin di AWS IAM Identity Center Panduan Pengguna.
2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

   Untuk petunjuk, lihat Menambahkan grup di AWS IAM Identity Center Panduan Pengguna.

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

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

Untuk mengakses AWS Elemental MediaStore

- Masuk keAWS 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, lihatMenyiapkan 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, lihatAWS ElementalMediaStoreHarga.

Topik

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 bernama`myContainer`dan folder bernama`mycontainer`karena 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 menggunakanAWS CloudFormationtemplat. ParameterAWS CloudFormationtemplat 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. BukaMediaStorekonsol dihttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerhalaman, pilihMembuat kontainer.
3. UntukKontainernama, masukkan nama untuk kontainer. Untuk informasi selengkapnya, lihat Aturan untuk nama kontainer.
4. PilihMembuat kontainer. AWS ElementalMediaStoremenambahkan wadah baru ke daftar kontainer. Awalnya, status kontainerMembuat, dan kemudian berubah menjadiAktif.

Untuk membuat wadah (AWS CLI)

- DiAWS 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. BukaMediaStorekonsol dihttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerhalaman, pilih nama kontainer.

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

   - ParameterObjekbagian, yang berisi daftar objek dan folder dalam wadah.
   - ParameterKontainerbagian kebijakan, yang menunjukkan kebijakan berbasis sumber daya yang terkait dengan wadah ini. Untuk informasi tentang kebijakan sumber daya, lihatKebijakan kontainer.

Untuk melihat detail untuk kontainer (AWS CLI)

- DiAWS 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://aaabbbcccdddee.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)

- BukaMediaStorekonsol dihttps://console.aws.amazon.com/mediastore/.

  ParameterKontainerhalaman muncul, daftar semua kontainer yang terkait dengan akun Anda.

Untuk melihat daftar kontainer (AWS CLI)

- DiAWS CLI, menggunakan`list-containers`perintah.

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

  Contoh berikut menunjukkan nilai kembali:

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

```
}
```

Menghapus kontainer

Anda dapat menghapus kontainer hanya jika tidak memiliki objek.

Untuk menghapus kontainer (konsol)

1. BukaMediaStorekonsol dihttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerhalaman, pilih opsi di sebelah kiri nama kontainer.
3. Pilih Delete (Hapus).

Untuk menghapus kontainer (AWS CLI)

- DiAWS 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 menentukankebijakan siklus hidup objekyang 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 atauAWS CLIuntuk melihat kebijakan berbasis sumber daya dari sebuah wadah.

Untuk melihat kebijakan kontainer (konsol)

1. BukaMediaStorekonsol dihttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerhalaman, pilih nama kontainer.

   Halaman detail kontainer akan muncul. Kebijakan ditampilkan dalamKebijakan kontainerbagian.

Untuk melihat kebijakan kontainer (AWS CLI)

- DiAWS 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. BukaMediaStorekonsol dihttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerhalaman, pilih nama kontainer.
3. Pilih Sunting kebijakan. Untuk contoh yang menunjukkan cara mengatur izin berbeda, lihatthe section called “Contoh kebijakan kontainer”.
4. Buat perubahan yang sesuai, lalu pilihSimpan.

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 ElementalMediaStoresecara 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 bisaedit kebijakanpada 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 semuaMediaStoreoperasi 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 mengaksesDeleteObject,GetObject, dan sebagainya (tapi tidakPutObject), dan dapat melakukan ini melalui HTTP atau HTTPS.

Jika Anda tidak mengecualikanPutObjectdari pernyataan kedua, pernyataan tidak akan valid (karena jika Anda menyertakanPutObjectAnda 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 ElementalMediaStoreoperasi 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.

Parameter`Condition`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, lihatOperator Kondisi Alamat IPdalam 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 denganMediaStoredan secara selektif mengizinkan akses lintas asal keMediaStoresumber daya.

> **Note**
>
> Jika Anda menggunakan AmazonCloudFrontuntuk mendistribusikan konten dari kontainer yang memiliki kebijakan CORS, pastikan untukmengkonfigurasi 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 ElementalMediaStorekonsol, atau pemrograman menggunakanMediaStoreREST API dan AWS SDK.

Topik

Skenario kasus penggunaan CORS

Berikut adalah contoh skenario untuk menggunakan CORS:

- Skenario 1: Misalkan Anda mendistribusikan video streaming langsung di AWS ElementalMediaStorewadah bernamaLiveVideo. Pengguna Anda memuat titik akhir videohttp://livevideo.mediastore.ap-southeast-2.amazonaws.comdari asal tertentu

sepertiwww.example.com. Anda ingin menggunakanJavaScriptpemutar video untuk mengakses video yang berasal dari kontainer ini melalui unauthenticatedGETdanPUTpermintaan. Browser biasanya akan memblokirJavaScriptdari mengizinkan permintaan tersebut, tetapi Anda dapat menetapkan kebijakan CORS pada kontainer Anda untuk secara eksplisit mengaktifkan permintaan ini dariwww.example.com.

- Skenario 2: Misalkan Anda ingin meng-host live stream yang sama seperti dalam Skenario 1 dari AndaMediaStorekontainer, 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 AWS ElementalMediaStorekontainer. 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. BukaMediaStorekonsol dihttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerHalaman, pilih nama wadah yang ingin Anda buat kebijakan CORS.

   Halaman detail akan muncul.
3. DiKebijakan CORSbagian, pilihBuat kebijakan CORS.
4. Masukkan kebijakan dalam format JSON, lalu pilihSimpan.

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. BukaMediaStorekonsol dihttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerHalaman, pilih nama kontainer yang ingin Anda lihat kebijakan CORS.

   Halaman rincian kontainer muncul, dengan kebijakan CORS diKebijakan CORSbagian.

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. BukaMediaStorekonsol dihttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerHalaman, pilih nama kontainer yang ingin Anda edit kebijakan CORS.

   Halaman detail akan muncul.
3. DiKebijakan CORSbagian, pilihMengedit kebijakan CORS.
4. Buat perubahan pada kebijakan, dan kemudian pilihSimpan.

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. BukaMediaStorekonsol dihttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerHalaman, pilih nama kontainer yang ingin Anda hapus kebijakan CORS.

   Halaman detail akan muncul.
3. DiKebijakan CORSbagian, pilihMenghapus kebijakan CORS.
4. MemiilihLanjutkanuntuk mengkonfirmasi, dan kemudian memilihSimpan.

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 ElementalMediaStoretidak 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`AllowedOrigins`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`AllowedMethods`elemen yang sama`CORSRule`.

d. Untuk permintaan preflight, jika permintaan mencakup header `Access-Control-Request-Headers`, verifikasikan 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

darihttps://www.example.com, dan layanan hanya merespon permintaan HTTP GET dan HTTP HEAD darihttps://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 ElementalMediaStoremenghapus objek. Anda dapat menghapus objek setelah tidak lagi diperlukan untuk menghemat biaya penyimpanan.

Anda juga dapat menentukanMediaStoreharus 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": "Footboll/index*.m3u8"} ],
                "seconds_since_create": [
```

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

Aturan data transien memiliki tiga bagian:

- `path`: Selalu diatur kewildcard. 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.m3us8, 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 memancarkanCloudWatchacara 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 untukMediaStoreuntuk 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 10`days_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 untukMediaStoreuntuk 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 ElementalMediaStoreuntuk menghapus objek yang disimpan dalam`LiveEvents/Football`folder dan`LiveEvents/Baseball`folder setelah mereka lebih tua dari 28 hari.
- Aturan kedua menginstruksikan layanan untuk menghapus objek yang disimpan dalam`LiveEvents/AwardsShow`folder ketika mereka berusia 15 hari atau lebih.
- Aturan ketiga menginstruksikan layanan untuk menghapus objek yang disimpan di mana saja di`LiveEvents`kontainer setelah mereka lebih tua dari 40 hari. Aturan ini berlaku untuk objek yang disimpan langsung di`LiveEvents`kontainer, serta benda-benda yang disimpan dalam salah satu dari empat subfolder kontainer.
- Aturan keempat menginstruksikan layanan untuk menghapus objek di`Football`folder yang cocok dengan pola`*.ts`setelah mereka lebih tua dari 20 hari.
- Aturan kelima menginstruksikan layanan untuk menghapus objek di`Football`folder yang cocok dengan pola`index*.m3u8`setelah mereka lebih tua dari 15 detik. MediaStoremenghapus file-file ini 16 detik setelah mereka ditempatkan dalam wadah.
- Aturan keenam menginstruksikan layanan untuk memindahkan objek di`Program`folder ke kelas penyimpanan IA setelah mereka berusia 30 hari.

Untuk contoh kebijakan siklus hidup objek lainnya, lihatKebijakan 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, lihatKomponen 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 10`days_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. BukaMediaStorekonsolhttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerHalaman, pilih nama kontainer yang ingin Anda buat kebijakan siklus hidup objek.

   Halaman detail akan muncul.
3. DiKebijakan siklus aktif objekbagian, pilihMembuat kebijakan siklus aktif objek.
4. Masukkan kebijakan dalam format JSON, lalu pilihSimpan.

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, gunakanput-lifecycle-policyperintah:

```
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. BukaMediaStorekonsolhttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerHalaman, pilih nama kontainer yang ingin Anda lihat kebijakan siklus hidup objek.

   Halaman rincian kontainer muncul, dengan kebijakan siklus hidup objek diKebijakan siklus aktif objekbagian.

Untuk melihat kebijakan siklus hidup objek (AWS CLI)

- DiAWS CLI, gunakanget-lifecycle-policyperintah:

```
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. BukaMediaStorekonsolhttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerHalaman, pilih nama kontainer yang ingin Anda edit kebijakan siklus hidup objek.

   Halaman detail akan muncul.
3. DiKebijakan siklus aktif objekbagian, pilihEdit kebijakan siklus aktif objek.
4. Buat kebijakan, dan pilihSimpan.

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, gunakanput-lifecycle-policyperintah:

```
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. BukaMediaStorekonsolhttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerHalaman, pilih nama kontainer yang ingin Anda hapus kebijakan siklus hidup objek.

   Halaman detail akan muncul.
3. DiKebijakan siklus aktif objekbagian, pilihMenghapus kebijakan siklus hidup.
4. MemiilihLanjutkanuntuk mengkonfirmasi, dan kemudian memilihSimpan.

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 bahwaMediaStoremenghapus objek yang cocok dengan semua kriteria berikut:

- Objek ditambahkan ke wadah setelah kebijakan menjadi efektif.
- Objek disimpan dalam`Football`folder.
- Objek memiliki ekstensi file`m3u8`.
- 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 bahwaMediaStoremenghapus objek yang cocok dengan semua kriteria berikut:

- Objek disimpan dalam`Program`pelipat
- 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 bahwaMediaStorememindahkan 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.

Parameterdays_since_createbidang 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 bahwaMediaStoremelakukan hal berikut:

- Pindahkan objek yang disimpan dalamAwardsShowfolder ke kelas penyimpanan akses jarang (IA) setelah 30 hari
- Hapus objek yang memiliki ekstensi filem3u8dan disimpan dalamFootballfolder setelah 20 detik
- Hapus objek yang disimpan dalamAprilfolder setelah 10 hari
- Hapus objek yang memiliki ekstensi fileltsdan disimpan dalamProgramfolder 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 menentukannyaMediaStoremenghapus semua objek dalam wadah, termasuk folder dan subfolder, 1 hari setelah ditambahkan ke wadah. Jika kontainer memegang objek apa pun sebelum kebijakan ini diterapkan,MediaStoremenghapus 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, lihatMediaStore 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 dimensi`ObjectGroupName=rule1` dan yang kedua memiliki dimensi`ObjectGroupName=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, lihatContoh 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 atauAWS 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, lihatContoh 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 ElementalMediaStoremenyimpan objek dalam folder. Jika folder tidak ada, layanan membuatnya, dan kemudian menyimpan objek dalam folder.

Misalnya, Anda memiliki sebuah kontainer dengan nama`movies`, dan Anda mengunggah file bernama`law.ts`dengan jalan`premium/canada`. AWS ElementalMediaStoremenyimpan objek di subfolder canada di bawah premium folder. Jika tidak ada folder, layanan akan membuat kedua`premium`folder dan`canada`subfolder, dan kemudian menyimpan objek Anda di`canada`subfolder. Jika Anda hanya menentukan wadah`movies`(tanpa jalan), layanan menyimpan objek langsung dalam wadah.

AWS ElementalMediaStoresecara 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`. 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.

Topik

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 ElementalMediaStoremenyimpan 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 ElementalMediaStoresecara 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 dapatmenambahkan kebijakan siklus hidup objekuntuk 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, lihatFolder 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 `kestreaming`. Anda dapat mengatur nilai header ini saat Andamengunggah objek menggunakan API. Jika Anda tidak menentukan header ini dalam permintaan Anda,MediaStorememberikan nilai default`standard`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. BukaMediaStorekonsol dihttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerhalaman, memilih nama wadah. Panel rincian untuk wadah muncul.
3. PilihMeng-unggah objek.
4. UntukJalur 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. DiObjekbagian, pilihJelajahi.
6. Arahkan ke folder yang sesuai, dan pilih satu objek untuk diunggah.
7. PilihBuka, dan kemudian pilihUnggah.

> **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)

- DiAWS 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 mengeditcontent-type,cache-control, ataupath.

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

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 menggunakanAWS CLIuntuk 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. BukaMediaStorekonsol dihttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerhalaman, 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. BukaMediaStorekonsol dihttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerhalaman, 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)

- DiAWS 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 padaseconds_since_createaturan tidak termasuk dalam`list-items`stanggapan.

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://
aaabbbcccddddee.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 ElementalMediaStoremenyimpan rincian seperti tanggal modifikasi, panjang konten, eTag (tag entitas), dan jenis konten. Untuk mempelajari bagaimana metadata objek digunakan, lihatMediaStoreinteraksi dengan cache HTTP.

Untuk melihat detail objek (konsol)

1. BukaMediaStorekonsol dihttps://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://
  aaabbbcccdddee.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": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
  }
  ```

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 dihttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerhalaman, memilih nama kontainer yang memiliki objek yang ingin Anda unduh.

3. Jika objek yang ingin Anda unduh ada di folder, lanjutkan memilih nama folder sampai Anda melihat objek.
4. Pilih nama objek.
5. PadaObjekhalaman detail, pilihUnduh.

Untuk mengunduh objek (AWS CLI)

- DiAWS CLI, gunakan`get-object`perintah:

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

Contoh berikut menunjukkan nilai kembali:

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

Untuk men-download bagian dari sebuah objek (AWS CLI)

- DiAWS CLI, gunakan`get-object`perintah, dan menentukan rentang.

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

Contoh berikut menunjukkan nilai kembali:

```
{
    "StatusCode": 206,
    "ContentRange": "bytes 0-100/2307346",
    "ContentLength": "101",
    "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
```

```
    "ContentType": "image/jpeg",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
}
```

Penghapusan objek

AWS ElementalMediaStoremenawarkan pilihan yang berbeda untuk menghapus objek dari kontainer:

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

Menghapus objek

Anda dapat menghapus objek secara terpisah menggunakan konsol atauAWS CLI. Atau, Anda dapatmenambahkan kebijakan siklus hidup objekuntuk secara otomatis menghapus objek setelah mereka mencapai usia tertentu dalam wadah, atau Anda dapatmengosongkan wadahuntuk 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`canada`subfolder berisi satu file bernama`law.ts`. Jika Anda menghapus file`law.ts`, layanan menghapus kedua`premium`dan`canada`folder.

Untuk menghapus objek (konsol)

1. BukaMediaStorekonsol dihttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerhalaman, 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)

- DiAWS 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 menambahkankebijakan siklus hidup objekuntuk secara otomatis menghapus objek setelah mereka mencapai usia tertentu dalam wadah, atau Anda dapatmenghapus objek secara individual.

Untuk mengosongkan kontainer (konsol)

1. BukaMediaStorekonsol dihttps://console.aws.amazon.com/mediastore/.
2. PadaKontainerhalaman, memilih opsi untuk wadah yang ingin Anda kosongkan.
3. PilihKontainer kosong. Sebuah pesan konfirmasi akan muncul.
4. Konfirmasikan bahwa Anda ingin mengosongkan kontainer dengan memasukkan nama kontainer ke dalam kolom teks, lalu pilihKosong.

Keamanan di AWS Elemental MediaStore

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat 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 kamu. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan cloud dan keamanan dalam cloud:

- Keamanan Cloud — AWS Bertanggung jawab untuk melindungi infrastruktur yang berjalan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari AWS Program Kepatuhan . Untuk mempelajari tentang program kepatuhan yang berlaku untuk AWS Elemental MediaStore, lihat AWS Layanan dalam Lingkup oleh Program Kepatuhan .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang 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 yang lain AWS layanan yang membantu Anda memantau dan mengamankan MediaStore sumber daya Anda.

Topik

Perlindungan data di AWS Elemental MediaStore

Bagian AWS model tanggung jawab bersama model berlaku untuk perlindungan data di AWS Elemental MediaStore. Seperti yang dijelaskan dalam model ini, AWS bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk menjaga kontrol atas konten Anda yang di-host di infrastruktur ini. Anda juga bertanggung jawab atas konfigurasi keamanan dan tugas manajemen untuk Layanan AWS yang Anda gunakan. Untuk informasi selengkapnya tentang privasi data, lihat Privasi Data FAQ. Untuk informasi tentang perlindungan data di Eropa, lihat AWS Model Tanggung Jawab Bersama dan posting GDPR blog di AWS Blog Keamanan.

Untuk tujuan perlindungan data, kami menyarankan Anda untuk melindungi Akun AWS kredensi dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan otentikasi multi-faktor (MFA) dengan setiap akun.
- GunakanSSL/TLSuntuk berkomunikasi dengan AWS sumber daya. Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang menggunakan CloudTrail jalur untuk menangkap AWS kegiatan, lihat Bekerja dengan CloudTrail jalan setapak di AWS CloudTrail Panduan Pengguna.
- Gunakan AWS solusi enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan FIPS 140-3 modul kriptografi yang divalidasi saat mengakses AWS melalui antarmuka baris perintah atauAPI, gunakan FIPS titik akhir. Untuk informasi selengkapnya tentang FIPS titik akhir yang tersedia, lihat Federal Information Processing Standard (FIPS) 140-3.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk ketika Anda bekerja dengan MediaStore atau lainnya Layanan AWS menggunakan konsol,API, AWS CLI, atau AWS SDKs. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan

atau log diagnostik. Jika Anda memberikan URL ke server eksternal, kami sangat menyarankan agar Anda tidak menyertakan informasi kredensil dalam URL untuk memvalidasi permintaan Anda ke server tersebut.

Enkripsi data

MediaStore mengenkripsi kontainer dan objek saat istirahat menggunakan standar industri AES -256 algoritma. 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 DenganCORS, 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

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya. IAMadministrator mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya. MediaStore IAMadalah sebuah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- Audiens
- Mengautentikasi dengan identitas
- Mengelola akses menggunakan kebijakan
- Bagaimana AWS Elemental MediaStore bekerja dengan IAM
- Contoh kebijakan berbasis identitas untuk Elemental AWS MediaStore
- Memecahkan masalah Identitas dan AWS akses Elemental MediaStore

Audiens

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

Pengguna layanan — Jika Anda menggunakan MediaStore layanan untuk melakukan pekerjaan Anda, maka 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 MediaStore, lihatMemecahkan masalah Identitas dan AWS akses Elemental MediaStore .

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. Anda kemudian harus mengirimkan permintaan ke IAM administrator Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep dasarIAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakannya IAM MediaStore, lihatBagaimana AWS Elemental MediaStore bekerja dengan IAM.

IAMadministrator - Jika Anda seorang IAM administrator, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses MediaStore. Untuk melihat contoh kebijakan MediaStore berbasis identitas yang dapat Anda gunakan, lihat. IAM Contoh kebijakan berbasis identitas untuk Elemental AWS MediaStore

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai IAM pengguna, atau dengan mengambil IAM peran.

Anda dapat masuk ke AWS sebagai identitas federasi dengan menggunakan kredensi yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (Pusat IAM Identitas), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas federasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan IAM peran. Saat Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Tergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau AWS portal akses. Untuk informasi lebih lanjut tentang masuk AWS, lihat Cara masuk ke Akun AWSdi AWS Sign-In Panduan Pengguna.

Jika Anda mengakses AWS secara terprogram, AWS menyediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensil Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang menggunakan metode yang disarankan untuk menandatangani permintaan sendiri, lihat Menandatangani AWS APIpermintaan di Panduan IAM Pengguna.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat Autentikasi multi-faktor di AWS IAM Identity Center Panduan Pengguna dan Menggunakan otentikasi multi-faktor () MFA di AWS di Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut Akun AWS pengguna root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar 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, lihat Tugas yang memerlukan kredensi pengguna root di IAMPanduan Pengguna.

Identitas gabungan

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, AWS Directory Service, direktori Pusat Identitas, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensi yang disediakan melalui sumber identitas. Ketika akses identitas federasi Akun AWS, mereka mengambil peran, dan peran memberikan kredensi sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat IAM Identitas, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua Akun AWS dan aplikasi. Untuk informasi tentang Pusat IAM Identitas, lihat Apa itu Pusat IAM Identitas? di AWS IAM Identity Center Panduan Pengguna.

Pengguna dan grup IAM

IAMPengguna adalah identitas di dalam Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, sebaiknya Anda mengandalkan kredensi sementara alih-alih membuat IAM pengguna yang memiliki kredensi jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensil jangka panjang dengan IAM pengguna, kami sarankan Anda memutar kunci akses. Untuk informasi selengkapnya, lihat Memutar kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensi jangka panjang di IAMPanduan Pengguna.

IAMGrup adalah identitas yang menentukan kumpulan IAM pengguna. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat memiliki grup bernama IAMAdminsdan memberikan izin grup tersebut untuk mengelola sumber dayaIAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari lebih lanjut, lihat Kapan membuat IAM pengguna (bukan peran) di Panduan IAM Pengguna.

IAMperan

IAMPeran adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Ini mirip dengan IAM pengguna, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil IAM peran sementara dalam AWS Management Console dengan beralih peran. Anda dapat mengambil peran dengan memanggil AWS CLI atau AWS APIoperasi atau dengan menggunakan kustomURL. Untuk informasi selengkapnya tentang metode penggunaan peran, lihat Menggunakan IAM peran di Panduan IAM Pengguna.

IAMperan dengan kredensi sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat Membuat peran untuk Penyedia Identitas pihak ketiga di Panduan IAM Pengguna. Jika Anda menggunakan Pusat IAM Identitas, Anda mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah diautentikasi, Pusat IAM Identitas mengkorelasikan izin yang disetel ke peran. IAM Untuk informasi tentang set izin, lihat Set izin di AWS IAM Identity Center Panduan Pengguna.
- Izin IAM pengguna sementara — IAM Pengguna atau peran dapat mengambil IAM peran untuk sementara mengambil izin yang berbeda untuk tugas tertentu.
- Akses lintas akun — Anda dapat menggunakan IAM peran untuk memungkinkan seseorang (prinsipal tepercaya) di akun lain mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM Panduan Pengguna. IAM
- Akses lintas layanan - Beberapa Layanan AWS menggunakan fitur di lain Layanan AWS. Misalnya, saat Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.
  - Teruskan sesi akses (FAS) — Saat Anda menggunakan IAM pengguna atau peran untuk melakukan tindakan AWS Anda dianggap sebagai kepala sekolah. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FASmenggunakan izin dari prinsipal yang memanggil Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. FASPermintaan hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk diselesaikan. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat FAS permintaan, lihat Meneruskan sesi akses.
  - Peran layanan — Peran layanan adalah IAMperan yang diasumsikan layanan untuk melakukan tindakan atas nama Anda. IAMAdministrator dapat membuat, memodifikasi, dan menghapus peran layanan dari dalamIAM. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke Layanan AWS di Panduan Pengguna IAM.

  - Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke Layanan AWS. Layanan dapat mengambil peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Akun AWS dan dimiliki oleh layanan. IAMAdministrator dapat melihat, tetapi tidak mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan IAM peran untuk mengelola kredensional sementara untuk aplikasi yang berjalan pada EC2 instance dan membuat AWS CLI atau AWS APIpermintaan. Ini lebih baik untuk menyimpan kunci akses dalam EC2 instance. Untuk menetapkan AWS peran ke sebuah EC2 instance dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instance berisi peran dan memungkinkan program yang berjalan pada EC2 instance untuk mendapatkan kredensi sementara. Untuk informasi selengkapnya, lihat Menggunakan IAM peran untuk memberikan izin ke aplikasi yang berjalan di EC2 instans Amazon di IAMPanduan Pengguna.

Untuk mempelajari apakah akan menggunakan IAM peran atau IAM pengguna, lihat Kapan membuat IAM peran (bukan pengguna) di Panduan IAM Pengguna.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses di AWS dengan membuat kebijakan dan melampirkannya AWS identitas atau sumber daya. Kebijakan adalah objek di AWS bahwa, ketika dikaitkan dengan identitas atau sumber daya, mendefinisikan izin mereka. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan di AWS sebagai JSON dokumen. Untuk informasi selengkapnya tentang struktur dan isi dokumen JSON kebijakan, lihat Ringkasan JSON kebijakan di Panduan IAM Pengguna.

Administrator dapat menggunakan AWS JSONkebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka butuhkan, IAM administrator dapat membuat IAM kebijakan. Administrator kemudian dapat menambahkan IAM kebijakan ke peran, dan pengguna dapat mengambil peran.

IAMkebijakan menentukan izin untuk tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasi. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan

`iam:GetRole`. Pengguna dengan kebijakan tersebut dapat memperoleh informasi peran dari AWS Management Console, AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan JSON izin yang dapat Anda lampirkan ke identitas, seperti pengguna, grup IAM pengguna, atau peran. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Membuat IAM kebijakan di Panduan Pengguna. IAM

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran di Akun AWS. Kebijakan terkelola meliputi AWS kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan sebaris, lihat Memilih antara kebijakan terkelola dan kebijakan sebaris di IAMPanduan Pengguna.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen JSON kebijakan yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan IAM peran 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 tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau Layanan AWS.

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

Daftar kontrol akses (ACLs)

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLsmirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan. JSON

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukungACLs. Untuk mempelajari selengkapnyaACLs, lihat Ikhtisar daftar kontrol akses (ACL) di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- Batas izin — Batas izin adalah fitur lanjutan tempat Anda menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas (pengguna atau peran). IAM IAM Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan 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 akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batas izin, lihat Batas izin untuk IAM entitas di IAMPanduan Pengguna.
- Kebijakan kontrol layanan (SCPs) — SCPs adalah JSON kebijakan yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations adalah layanan untuk mengelompokkan dan mengelola beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur dalam suatu organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCPs) ke salah satu atau semua akun Anda. SCPMembatasi izin untuk entitas di akun anggota, termasuk masing-masing Pengguna root akun AWS. Untuk informasi selengkapnya tentang Organizations danSCPs, lihat Kebijakan kontrol layanan di AWS Organizations Panduan Pengguna.
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan secara tegas dalam salah satu kebijakan ini membatalkan izin. Untuk informasi selengkapnya, lihat Kebijakan sesi di Panduan IAM Pengguna.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari caranya AWS menentukan apakah akan mengizinkan

permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan IAM Pengguna.

Bagaimana AWS Elemental MediaStore bekerja dengan IAM

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

IAMfitur yang dapat Anda gunakan dengan AWS Elemental MediaStore

IAMfitur	MediaStore dukungan
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Ya
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
kunci-kunci persyaratan kebijakan (spesifik layanan)	Ya
ACLs	Tidak
ABAC(tag dalam kebijakan)	Parsial
Kredensial sementara	Ya
Izin prinsipal	Ya
Peran layanan	Ya
Peran terkait layanan	Tidak

Untuk mendapatkan pandangan tingkat tinggi tentang bagaimana MediaStore dan lainnya AWS layanan bekerja dengan sebagian besar IAM fitur, lihat AWS layanan yang bekerja dengan IAM dalam Panduan IAM Pengguna.

Kebijakan berbasis identitas untuk MediaStore

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan JSON izin yang dapat Anda lampirkan ke identitas, seperti pengguna, grup IAM pengguna, atau peran. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Membuat IAM kebijakan di Panduan Pengguna. IAM

Dengan kebijakan IAM berbasis identitas, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak serta kondisi di mana tindakan diizinkan atau ditolak. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam JSON kebijakan, lihat referensi elemen IAM JSON kebijakan di Panduan IAM Pengguna.

Contoh kebijakan berbasis identitas untuk MediaStore

Untuk melihat contoh kebijakan MediaStore berbasis identitas, lihat. Contoh kebijakan berbasis identitas untuk Elemental AWS MediaStore

Kebijakan berbasis sumber daya dalam MediaStore

Mendukung kebijakan berbasis sumber daya: Ya

Kebijakan berbasis sumber daya adalah dokumen JSON kebijakan yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan IAM peran 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 tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau Layanan AWS.

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan seluruh akun atau IAM entitas di akun lain sebagai prinsipal dalam kebijakan berbasis sumber daya. Menambahkan prinsipal akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan. Ketika kepala sekolah dan sumber daya berbeda Akun AWS, IAM administrator di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya.

Mereka memberikan izin dengan melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses ke prinsipal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun IAM di Panduan IAM Pengguna.

> **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 AWS JSONkebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

`Action`Elemen JSON kebijakan menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan yang terkait AWS APIoperasi. Ada beberapa pengecualian, seperti tindakan khusus izin yang tidak memiliki operasi yang cocok. API Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi 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 tersebut dengan koma.

```
"Action": [
```

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

Untuk melihat contoh kebijakan MediaStore berbasis identitas, lihat. Contoh kebijakan berbasis identitas untuk Elemental AWS MediaStore

Sumber daya kebijakan untuk MediaStore

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan AWS JSONkebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Elemen `Resource` JSON kebijakan menentukan objek atau objek yang tindakan tersebut berlaku. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Sebagai praktik terbaik, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk 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 menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

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

Untuk melihat daftar jenis MediaStore sumber daya dan jenisnyaARNs, lihat Sumber daya yang ditentukan oleh AWS Elemental MediaStore 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 yang berikutARN:

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

Untuk informasi selengkapnya tentang formatARNs, lihat Amazon Resource Names (ARNs) dan AWS Ruang Nama Layanan.

Misalnya, untuk menentukan `AwardsShow` wadah dalam pernyataan Anda, gunakan yang berikut iniARN:

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

Kunci kondisi kebijakan untuk MediaStore

Mendukung kunci kondisi kebijakan khusus layanan: Ya

Administrator dapat menggunakan AWS JSONkebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

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

Jika Anda menentukan beberapa `Condition` elemen dalam pernyataan, atau beberapa kunci dalam satu `Condition` elemen, AWS mengevaluasi mereka menggunakan `AND` operasi logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Misalnya, Anda dapat memberikan izin IAM pengguna untuk mengakses sumber daya hanya jika ditandai dengan nama IAM pengguna mereka. Untuk informasi selengkapnya, lihat elemen IAM kebijakan: variabel dan tag di Panduan IAM Pengguna.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua AWS kunci kondisi global, lihat AWS kunci konteks kondisi global di Panduan IAM Pengguna.

Untuk melihat daftar kunci MediaStore kondisi, lihat Kunci kondisi untuk AWS Elemental MediaStore dalam 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 Elemental AWS MediaStore

ACLsdi MediaStore

MendukungACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLsmirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan. JSON

ABACdengan MediaStore

Mendukung ABAC (tag dalam kebijakan): Sebagian

Attribute-based access control (ABAC) adalah strategi otorisasi yang mendefinisikan izin berdasarkan atribut. Masuk AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke IAM entitas (pengguna atau peran) dan ke banyak AWS sumber daya. Menandai entitas dan sumber daya adalah langkah pertama dari. ABAC Kemudian Anda merancang ABAC kebijakan untuk mengizinkan operasi ketika tag prinsipal cocok dengan tag pada sumber daya yang mereka coba akses.

ABACmembantu dalam lingkungan yang berkembang pesat dan membantu dengan situasi di mana manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tag di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

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

Untuk informasi lebih lanjut tentangABAC, lihat Apa ituABAC? dalam IAMUser Guide. Untuk melihat tutorial dengan langkah-langkah penyiapanABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) di IAMPanduan Pengguna.

Menggunakan kredensi sementara dengan MediaStore

Mendukung kredensi sementara: Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensi sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensi sementara, lihat Layanan AWS yang bekerja dengan IAM dalam Panduan IAM Pengguna.

Anda menggunakan kredensi sementara jika Anda masuk ke AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan tautan single sign-on (SSO) perusahaan Anda, proses itu secara

otomatis membuat kredensi sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang beralih peran, lihat Beralih ke peran (konsol) di Panduan IAM Pengguna.

Anda dapat secara manual membuat kredensi sementara menggunakan AWS CLI atau AWS API. Anda kemudian dapat menggunakan kredensi sementara tersebut untuk mengakses AWS. AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensi keamanan sementara di. IAM

Izin utama lintas layanan untuk MediaStore

Mendukung sesi akses maju (FAS): Ya

Saat Anda menggunakan IAM pengguna atau peran untuk melakukan tindakan AWS Anda dianggap sebagai kepala sekolah. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FASmenggunakan izin dari prinsipal yang memanggil Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. FASPermintaan hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk diselesaikan. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat FAS permintaan, lihat Meneruskan sesi akses.

Peran layanan untuk MediaStore

Mendukung peran layanan: Ya

Peran layanan adalah IAMperan yang diasumsikan layanan untuk melakukan tindakan atas nama Anda. IAMAdministrator dapat membuat, memodifikasi, dan menghapus peran layanan dari dalamIAM. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke Layanan AWS di 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 terkait layanan: Tidak

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke Layanan AWS. Layanan dapat mengambil peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Akun AWS dan dimiliki oleh layanan. IAMAdministrator dapat melihat, tetapi tidak mengedit izin untuk peran terkait layanan.

Untuk detail tentang membuat atau mengelola peran terkait layanan, lihat AWS layanan yang bekerja dengan IAM. Cari layanan dalam tabel yang memiliki `Yes` di kolom Peran terkait layanan. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Contoh kebijakan berbasis identitas untuk Elemental AWS MediaStore

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

Untuk mempelajari cara membuat kebijakan IAM berbasis identitas menggunakan contoh dokumen kebijakan ini, lihat Membuat JSON IAM kebijakan di Panduan Pengguna. IAM

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh MediaStore, termasuk format ARNs untuk setiap jenis sumber daya, lihat Kunci tindakan, sumber daya, dan kondisi untuk AWS Elemental MediaStore dalam Referensi Otorisasi Layanan.

Topik

- Praktik terbaik kebijakan
- Menggunakan konsol MediaStore
- Mengizinkan pengguna 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 dapat menimbulkan biaya untuk

Anda Akun AWS. Saat Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi berikut:

- Memulai dengan AWS kebijakan terkelola dan beralih ke izin hak istimewa terkecil — Untuk memulai pemberian izin kepada pengguna dan beban kerja Anda, gunakan AWS kebijakan terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan mendefinisikan AWS kebijakan yang dikelola pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, silakan lihat AWS kebijakan terkelola atau AWS kebijakan terkelola untuk fungsi pekerjaan di Panduan IAM Pengguna.
- Menerapkan izin hak istimewa paling sedikit — Saat Anda menetapkan izin dengan IAM kebijakan, berikan hanya izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang penggunaan IAM untuk menerapkan izin, lihat Kebijakan dan izin IAM di IAM Panduan Pengguna.
- Gunakan ketentuan dalam IAM kebijakan untuk membatasi akses lebih lanjut — Anda dapat menambahkan kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Misalnya, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakanSSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui tindakan tertentu Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat elemen IAM JSON kebijakan: Kondisi dalam Panduan IAM Pengguna.
- Gunakan IAM Access Analyzer untuk memvalidasi IAM kebijakan Anda guna memastikan izin yang aman dan fungsional — IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan mematuhi bahasa IAM kebijakan () JSON dan praktik terbaik. IAM IAMAccess Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan IAM Access Analyzer di IAMPanduan Pengguna.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan IAM pengguna atau pengguna root di Akun AWS, nyalakan MFA untuk keamanan tambahan. Untuk meminta MFA kapan API operasi dipanggil, tambahkan MFA kondisi ke kebijakan Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi API akses MFA yang dilindungi di IAMPanduan Pengguna.

Untuk informasi selengkapnya tentang praktik terbaik diIAM, lihat Praktik terbaik keamanan IAM di Panduan IAM Pengguna.

Menggunakan konsol MediaStore

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

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang cocok dengan API operasi yang mereka coba lakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan MediaStore konsol, lampirkan MediaStore *ConsoleAccess* juga *ReadOnly* AWS kebijakan yang dikelola untuk entitas. Untuk informasi selengkapnya, lihat Menambahkan izin ke pengguna di Panduan IAM Pengguna.

Mengizinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara Anda membuat kebijakan yang memungkinkan IAM pengguna melihat kebijakan sebaris dan terkelola yang dilampirkan pada identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau secara terprogram menggunakan AWS CLI atau AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
```

```
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Memecahkan masalah Identitas dan AWS akses Elemental MediaStore

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

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 kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

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

```
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 AWS administrator. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

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.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika IAM pengguna bernama `marymajor` mencoba menggunakan konsol 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 AWS administrator. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

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

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis

sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mempelajari apakah MediaStore mendukung fitur-fitur ini, lihatBagaimana AWS Elemental MediaStore bekerja dengan IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh Akun AWS yang Anda miliki, lihat Menyediakan akses ke IAM pengguna di pengguna lain Akun AWS yang Anda miliki di Panduan IAM Pengguna.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda ke pihak ketiga Akun AWS, lihat Menyediakan akses ke Akun AWS dimiliki oleh pihak ketiga dalam Panduan IAM Pengguna.
- Untuk mempelajari cara menyediakan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna yang diautentikasi secara eksternal (federasi identitas) di Panduan Pengguna. IAM
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM Panduan Pengguna. IAM

Penebangan dan pemantauan di AWS Elemental MediaStore

Bagian ini memberikan gambaran umum tentang opsi untuk pencatatan dan pemantauan di AWS Elemental MediaStore untuk tujuan keamanan. Untuk informasi selengkapnya tentang pencatatan dan pemantauan MediaStore, lihatPemantauan dan penandaan di AWS Elemental MediaStore.

Pemantauan merupakan bagian penting dari menjaga keandalan, ketersediaan, dan kinerja AWS Elemental MediaStore dan kamu AWS solusi. Anda harus mengumpulkan data pemantauan dari semua bagian AWS solusi sehingga Anda dapat lebih mudah men-debug kegagalan multi-point 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 SNS topik Amazon 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.

AWS CloudTrail log

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 AWS lingkungan Anda dan kemudian membuat rekomendasi ketika ada peluang untuk menghemat uang, meningkatkan ketersediaan dan kinerja sistem, atau membantu menutup kesenjangan keamanan. Semua AWS pelanggan memiliki akses ke lima cek Trusted Advisor. Pelanggan dengan paket dukungan Bisnis atau Perusahaan dapat melihat semua Trusted Advisor cek.

Untuk informasi selengkapnya, silakan lihat AWS Trusted Advisor.

Validasi kepatuhan untuk AWS Elemental MediaStore

Untuk mengetahui apakah Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat Layanan AWS dalam Lingkup oleh Program Kepatuhan dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat AWS Program Kepatuhan .

Anda dapat 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, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Panduan Memulai Cepat Keamanan dan Kepatuhan — Panduan penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar AWS yang berfokus pada keamanan dan kepatuhan.
- Arsitektur untuk HIPAA Keamanan dan Kepatuhan di Amazon Web Services — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat aplikasi HIPAA yang memenuhi syarat.

> **Note**
>
> Tidak semua Layanan AWS HIPAAmemenuhi syarat. Untuk informasi selengkapnya, lihat Referensi Layanan yang HIPAA Memenuhi Syarat.

- AWS Sumber Daya Kepatuhan — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- AWS Panduan Kepatuhan Pelanggan - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- Mengevaluasi Sumber Daya dengan Aturan di AWS Config Panduan Pengembang — The AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalam AWS. Security Hub menggunakan kontrol keamanan untuk mengevaluasi AWS sumber daya dan untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat Referensi kontrol Security Hub.
- Amazon GuardDuty - Ini Layanan AWS Mendeteksi potensi ancaman terhadap Akun AWS, beban kerja, kontainer, dan data dengan memantau lingkungan Anda untuk aktivitas yang mencurigakan dan berbahaya. GuardDuty dapat membantu Anda mengatasi berbagai persyaratan kepatuhan, seperti PCIDSS, dengan memenuhi persyaratan deteksi intrusi yang diamanatkan oleh kerangka kerja kepatuhan tertentu.
- AWS Audit Manager— Ini Layanan AWS membantu Anda terus mengaudit AWS penggunaan untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Ketahanan dalam Elemental AWS MediaStore

Bagian AWS Infrastruktur global dibangun di sekitar Wilayah AWS dan Availability Zone. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data

yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi lebih lanjut tentang Wilayah AWS dan Availability Zone, lihat AWS Infrastruktur Global.

Selain AWS Infrastruktur 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 AWS keamanan jaringan global. Untuk informasi tentang AWS Layanan keamanan dan bagaimana AWS melindungi infrastruktur, lihat AWS Keamanan Cloud. Untuk mendesain Anda AWS lingkungan menggunakan praktik terbaik untuk keamanan infrastruktur, lihat Perlindungan Infrastruktur di Pilar Keamanan AWS Kerangka Kerja yang Diarsiteksikan dengan Baik.

Anda menggunakan AWS APIpanggilan yang diterbitkan untuk mengakses MediaStore melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Transportasi (TLS). Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Suite cipher dengan kerahasiaan maju yang sempurna (PFS) seperti (Ephemeral Diffie-Hellman) atau DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani dengan menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan IAM prinsipal. Atau Anda dapat menggunakan AWS Security Token Service (AWS STS) untuk menghasilkan kredensi keamanan sementara untuk 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. Masuk AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang

membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan pemanggilan dapat dimanipulasi menggunakan izinnya untuk bertindak pada sumber daya pelanggan lain dengan cara yang seharusnya tidak dilakukannya kecuali bila memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data Anda untuk semua layanan dengan prinsipal layanan yang telah diberikan akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi `aws:SourceAccount`global `aws:SourceArn`dan dalam kebijakan sumber daya untuk membatasi izin yang MediaStore diberikan AWS 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 tersebut 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 penuh ARN sumber daya. Jika Anda tidak tahu sumber daya penuh ARN atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks `aws:SourceArn` global dengan karakter wildcard (*) untuk bagian yang tidak diketahui dari file. ARN Misalnya, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Jika `aws:SourceArn` nilainya tidak berisi ID akun, seperti bucket Amazon S3ARN, 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": "servicename.amazonaws.com"
    },
    "Action": "servicename:ActionName",
    "Resource": [
      "arn:aws:servicename:::ResourceName/*"
```

```
    ],
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:servicename:*: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 danAWS solusi Anda lainnya. AWSmenyediakan 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 memantauAWS sumber daya Anda dan aplikasi yang Anda jalankanAWS 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 dalamAWS 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 diAWS 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 mengaturAWS sumber daya Anda diAWS Management Console, membuat laporan penggunaan dan penagihan di semuaAWS sumber daya Anda, dan memfilter sumber daya selama aktivitas otomatisasi infrastruktur.

Topik

Membuat log panggilan MediaStore API AWS Elemental denganAWS CloudTrail

AWS Elemental MediaStore terintegrasi denganAWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atauAWS 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 PanduanAWS CloudTrail Pengguna.

Topik

MediaStoreInformasi AWS Elemental di CloudTrail

CloudTrail diaktifkan diAWS akun Anda saat Anda membuat akun. Saat aktivitas peristiwa yang didukung terjadi di AWS Elemental MediaStore, aktivitas tersebut dicatat di CloudTrail peristiwa bersama dengan peristiwaAWS 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 mengonfigurasiAWS 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 kredenal pengguna root atau kredenal 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. Peristuwa 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":
 "MNCTGH4HRQJ27GRMBVDPIVHEP4LO2BN6MUVHBCPSHOAWNSOKSXCO24B2UEOBBND5DONRXTMFK3TOJ4G7AHWMESI",
    "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.

AWSmenyediakan 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 dariAWS 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 jumlah`PutObject` 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 dalamAWS 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

GunakanAWS 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 dalamAWS 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.) AWStidak 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)

- DalamAWS 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)

- DalamAWS 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 mengaktifkanAWS 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 the section called “Mengatur izin untuk CloudWatch”.
Anda mengirimkan permintaa `nDescribeContainer` API, tetapi respons menunjukkan bahwa`AccessLoggingEnabl ed` parameter memiliki nilai`False`. Selain itu, Anda tidak melihat CloudTrail acara apa pun untuk`MediaStoreAccessLogs` peran yang berhasil`DescribeL ogGroup` ,`CreateLog Group` ,`DescribeLogStream` , atau`CreateLogStream` panggilan.	Peran IAM tidak ada atau memiliki nama, izin, atau kebijakan kepercayaan yang salah.	Buat peran dengan nama, izin, dan kebijakan kepercayaan yang benar. Lihat the section called “Mengatur izin untuk CloudWatch”.
	Pencatatan akses tidak diaktifkan pada kontainer.	Aktifkan log akses untuk wadah. Lihat the section called “Mengaktifkan pengelogan akses”.
Di CloudTrail konsol, Anda melihat peristiwa dengan kesalahan akses ditolak yang terkait dengan`MediaStor eAccessLogs` peran tersebut.	Peran IAM tidak memiliki izin yang benar untuk AWS Elemental MediaStor e.	Perbarui peran IAM untuk memiliki izin yang benar dan kebijakan kepercayaan. Lihat the section called “Mengatur izin untuk CloudWatch”.

Gejala	Masalahnya Mungkin...	Coba ini...

CloudTrail Acara ini mungkin mencakup baris seperti berikut ini:
 	
`"eventSource": "logs.amazonaws.com",`
 	
`"errorCode": "AccessDenied",`
 	
`"errorMessage": "User: arn:aws:sts::111122223333:assumed-role/MediaStoreAccessLogs/MediaStoreAccessLogsSession is not authorized to perform: logs:DescribeLogGroups on resource: arn:aws:logs:us-west-2:111122223333:log-group::log-stream:",`		

Anda tidak melihat log untuk seluruh kontainer atau kontainer.	Akun Anda mungkin telah melampaui CloudWatch kuota untuk grup log per Wilayah. Lihat kuota untuk grup log di Panduan Pengguna Amazon CloudWatch Logs.	Di CloudWatch konsol, tentukan apakah akun Anda telah memenuhi CloudWatch kuota untuk grup log. Jika perlu, minta kenaikan kuota.

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 Panduan Pengguna Amazon CloudWatch Logs.	Meminta CloudWatch kenaikan kuota 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":
 "aaaAAA111bbbBBB222cccCCC333dddDDD444eeeEEE555fffFFF666gggGGG777hhhHHH888iiiIII999jjjJJJ",
  "ContainerName": "LiveEvents",
  "TotalTime": 147,
  "BytesReceived": 1572864,
  "BytesSent": 184,
  "ReceivedTime": "2018-12-13T12:22:06.245Z",
  "Operation": "PutObject",
  "ErrorCode": null,
  "Source": "192.0.2.3",
  "HTTPStatus": 200,
  "TurnAroundTime': 7,
  "ExpiresAt": "2018-12-13T12:22:36Z"
}
 {
  "Path": "/FootballMatch/West",
  "Requester": "arn:aws:iam::111122223333:user/maria-garcia",
```

```
  "AWSAccountId": "111122223333",
  "RequestID":
 "dddDDD444eeeEEE555fffFFF666gggGGG777hhhHHH888iiiIII999jjjJJJ000cccCCC333bbbBBB222aaaAAA",
  "ContainerName": "LiveEvents",
  "TotalTime": 3,
  "BytesReceived": 641354,
  "BytesSent": 163,
  "ReceivedTime": "2018-12-13T12:22:51.779Z",
  "Operation": "PutObject",
  "ErrorCode": "ValidationException",
  "Source": "198.51.100.15",
  "HTTPStatus": 400,
  "TurnAroundTime": 1,
  "ExpiresAt": null
}
```


Daftar berikut menjelaskan bidang catatan log:

AWSAccountId

IDAWS akun dari 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 usiatransient 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 mengotomatiskanAWS 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 dipecat 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 atauAWS 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. CloudWatchmembuat 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 keAWS 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/namespace.
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`	Jumlah total permintaan HTTP yang dibuat untuk MediaStore wadah, dipisahkan oleh jenis operasi (`Put`,,`Get`,`Delete`,`Describe`,`List`). Unit: Count (Jumlah) Dimensi yang valid: • Nama kontainer • Nama grup objek • Jenis permintaan

Metrik	Deskripsi
	Statistik yang valid: Sum
`4xxErrorCount`	Jumlah permintaan HTTP yang dibuat untuk MediaStore yang mengakibatkan kesalahan 4xx. Unit: Count (Jumlah) Dimensi yang valid: • Nama kontainer • Nama grup objek • Jenis permintaan Statistik yang valid: Sum
`5xxErrorCount`	Jumlah permintaan HTTP yang dibuat untuk MediaStore yang mengakibatkan kesalahan 5xx. Unit: Count (Jumlah) Dimensi yang valid: • Nama kontainer • Nama grup objek • Jenis permintaan Statistik yang valid: Sum

Metrik	Deskripsi
`BytesUploaded`	Jumlah byte yang diunggah untuk permintaan yang dibuat ke MediaStore wadah, di mana permintaan mencakup satu bodi. Unit: Bita Dimensi yang valid: • Nama kontainer • Nama grup objek 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
`BytesDownloaded`	Jumlah byte yang diunduh untuk permintaan yang dibuat ke MediaStore wadah, yang tanggapannya mencakup satu bodi. Unit: Bita Dimensi yang valid: • Nama kontainer • Nama grup objek 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

Metrik	Deskripsi
`TotalTime`	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 perspekti f server karena pengukuran yang dilakukan dari perspektif klien dipengaruhi oleh latensi jaringan. Unit: Milidetik Dimensi yang valid: • Nama kontainer • Nama grup objek • Jenis permintaan Rata-rata statistik yang valid, Min (sama dengan P0,0), Maks (sama dengan p100), setiap persentil antara p0,0 dan p100
`Turnaroun dTime`	Jumlah milidetik yang MediaStore dihabiskan untuk memproses permintaa n Anda. Nilai ini diukur dari waktu yang MediaStore menerima byte terakhir permintaan Anda, hingga saat mengirim byte pertama respons. Unit: Milidetik Dimensi yang valid: • Nama kontainer • Nama grup objek • Jenis permintaan Rata-rata statistik yang valid, Min (sama dengan P0,0), Maks (sama dengan p100), setiap persentil antara p0,0 dan p100

Metrik	Deskripsi
`ThrottleCount`	Jumlah permintaan HTTP yang dibuat untuk MediaStore yang dicekatkan. Unit: Count (Jumlah) Dimensi yang valid: • Nama kontainer • Nama grup objek • Jenis permintaan Statistik yang valid: Sum

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.

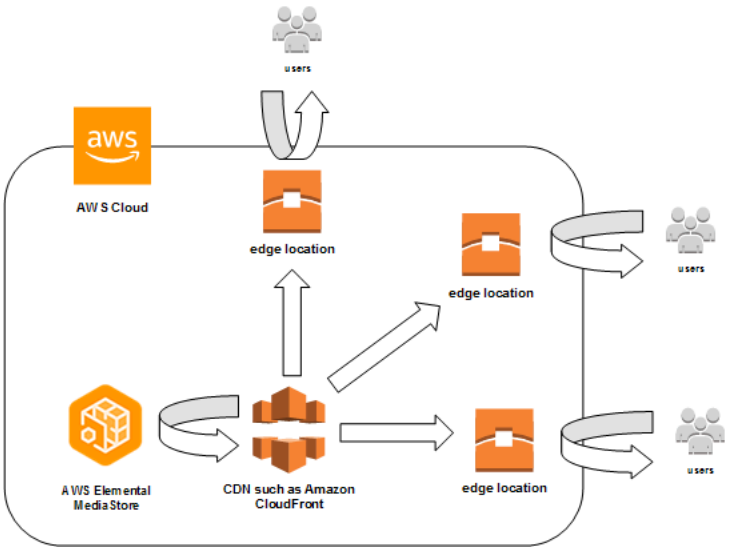
Mengelola tag

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 pengenal 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.

Menggunakan layanan ini dengan AWS SDK

AWS kit pengembangan perangkat lunak (SDKs) tersedia untuk banyak bahasa pemrograman populer. Masing-masing SDK menyediakanAPI, contoh kode, dan dokumentasi yang memudahkan pengembang untuk membangun aplikasi dalam bahasa pilihan mereka.

SDKdokumentasi	Contoh kode
AWS SDK for C++	AWS SDK for C++ contoh kode
AWS CLI	AWS CLI contoh kode
AWS SDK for Go	AWS SDK for Go contoh kode
AWS SDK for Java	AWS SDK for Java contoh kode
AWS SDK for JavaScript	AWS SDK for JavaScript contoh kode
AWS SDK for Kotlin	AWS SDK for Kotlin contoh kode
AWS SDK for .NET	AWS SDK for .NET contoh kode
AWS SDK for PHP	AWS SDK for PHP contoh kode
AWS Tools for PowerShell	Alat untuk contoh PowerShell kode
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) contoh kode
AWS SDK for Ruby	AWS SDK for Ruby contoh kode
AWS SDK for Rust	AWS SDK for Rust contoh kode
AWS SDK untuk SAP ABAP	AWS SDK untuk SAP ABAP contoh kode
AWS SDK for Swift	AWS SDK for Swift contoh kode

Ketersediaan contoh

Tidak dapat menemukan apa yang Anda butuhkan? Minta contoh kode menggunakan tautan Berikan umpan balik di bagian bawah halaman ini.

Contoh kode untuk MediaStore menggunakan AWS SDKs

Contoh kode berikut menunjukkan bagaimana menggunakan MediaStore dengan AWS kit pengembangan perangkat lunak (SDK).

Tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Untuk daftar lengkap AWS SDKpanduan pengembang dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Contoh kode

- Contoh dasar untuk MediaStore menggunakan AWS SDKs
  - Tindakan untuk MediaStore menggunakan AWS SDKs
    - Gunakan CreateContainer dengan AWS SDKatau CLI
    - Gunakan DeleteContainer dengan AWS SDKatau CLI
    - Gunakan DeleteObject dengan AWS SDKatau CLI
    - Gunakan DescribeContainer dengan AWS SDKatau CLI
    - Gunakan GetObject dengan AWS SDKatau CLI
    - Gunakan ListContainers dengan AWS SDKatau CLI
    - Gunakan PutObject dengan AWS SDKatau CLI

Contoh dasar untuk MediaStore menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan dasar-dasar AWS Elemental MediaStore dengan AWS SDKs.

Contoh

- Tindakan untuk MediaStore menggunakan AWS SDKs
  - Gunakan CreateContainer dengan AWS SDKatau CLI
  - Gunakan DeleteContainer dengan AWS SDKatau CLI
  - Gunakan DeleteObject dengan AWS SDKatau CLI

- Gunakan DescribeContainer dengan AWS SDKatau CLI
- Gunakan GetObject dengan AWS SDKatau CLI
- Gunakan ListContainers dengan AWS SDKatau CLI
- Gunakan PutObject dengan AWS SDKatau CLI

Tindakan untuk MediaStore menggunakan AWS SDKs

Contoh kode berikut menunjukkan bagaimana melakukan MediaStore tindakan individu dengan AWS SDKs. Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan instruksi untuk mengatur dan menjalankan kode.

Contoh berikut hanya mencakup tindakan yang paling umum digunakan. Untuk daftar lengkap, lihat AWS Elemental MediaStore APIReferensi.

Contoh

- Gunakan CreateContainer dengan AWS SDKatau CLI
- Gunakan DeleteContainer dengan AWS SDKatau CLI
- Gunakan DeleteObject dengan AWS SDKatau CLI
- Gunakan DescribeContainer dengan AWS SDKatau CLI
- Gunakan GetObject dengan AWS SDKatau CLI
- Gunakan ListContainers dengan AWS SDKatau CLI
- Gunakan PutObject dengan AWS SDKatau CLI

Gunakan **`CreateContainer`** dengan AWS SDKatau CLI

Contoh kode berikut menunjukkan cara menggunakan`CreateContainer`.

CLI

AWS CLI

Untuk membuat wadah

`create-container`Contoh berikut membuat wadah baru yang kosong.

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

Output:

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

Untuk informasi selengkapnya, lihat Membuat Kontainer di AWS Panduan MediaStore Pengguna Elemental.

- Untuk API detailnya, lihat CreateContainerdi AWS CLI Referensi Perintah.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di AWS Repositori Contoh Kode.

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

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

```
 */
public class CreateContainer {
    public static long sleepTime = 10;

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

                Usage:    <containerName>

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

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

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

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

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

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

```
            }

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

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

- Untuk API detailnya, lihat CreateContainerdi AWS SDK for Java 2.x APIReferensi.

Untuk daftar lengkap AWS SDKpanduan pengembang dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`DeleteContainer`** dengan AWS SDKatau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteContainer`.

CLI

AWS CLI

Untuk menghapus wadah

`delete-container`Contoh berikut menghapus wadah yang ditentukan. Anda dapat menghapus wadah hanya jika tidak memiliki objek.

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

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat Menghapus Kontainer di AWS Panduan MediaStore Pengguna Elemental.

- Untuk API detailnya, lihat DeleteContainerdi AWS CLI Referensi Perintah.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di AWS Repositori Contoh Kode.

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

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

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

                Usage:    <containerName>

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

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

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

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

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

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

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

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

- Untuk API detailnya, lihat DeleteContainerdi AWS SDK for Java 2.x APIReferensi.

Untuk daftar lengkap AWS SDKpanduan pengembang dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`DeleteObject`** dengan AWS SDKatau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteObject`.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di AWS Repositori Contoh Kode.

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

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

```
                Usage:    <completePath> <containerName>

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

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

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

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

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

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

            mediaStoreData.deleteObject(deleteObjectRequest);

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

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

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

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

- Untuk API detailnya, lihat DeleteObjectdi AWS SDK for Java 2.x APIReferensi.

Untuk daftar lengkap AWS SDKpanduan pengembang dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`DescribeContainer`** dengan AWS SDKatau CLI

Contoh kode berikut menunjukkan cara menggunakan`DescribeContainer`.

CLI

AWS CLI

Untuk melihat detail kontainer

`describe-container`Contoh berikut menampilkan rincian wadah yang ditentukan.

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

Output:

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

Untuk informasi selengkapnya, lihat Melihat Detail untuk Kontainer di AWS Panduan MediaStore Pengguna Elemental.

- Untuk API detailnya, lihat DescribeContainerdi AWS CLI Referensi Perintah.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di AWS Repositori Contoh Kode.

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

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
```

```
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeContainer {

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

                Usage:    <containerName>

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

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

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

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

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

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

```
            return containerResponse.container().status().toString();

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

- Untuk API detailnya, lihat DescribeContainerdi AWS SDK for Java 2.x APIReferensi.

Untuk daftar lengkap AWS SDKpanduan pengembang dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`GetObject`** dengan AWS SDKatau CLI

Contoh kode berikut menunjukkan cara menggunakan`GetObject`.

CLI

AWS CLI

Untuk mengunduh objek

`get-object`Contoh berikut men-download objek ke endpoint yang ditentukan.

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

Output:

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

Untuk mengunduh bagian dari suatu objek

`get-object`Contoh berikut mendownload sebagian objek ke titik akhir yang ditentukan.

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

Output:

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

Untuk informasi selengkapnya, lihat Mengunduh Objek di AWS Panduan MediaStore Pengguna Elemental.

- Untuk API detailnya, lihat GetObjectdi AWS CLI Referensi Perintah.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di AWS Repositori Contoh Kode.

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

```
import software.amazon.awssdk.services.mediastoredata.MediaStoreDataClient;
import software.amazon.awssdk.services.mediastoredata.model.GetObjectRequest;
import software.amazon.awssdk.services.mediastoredata.model.GetObjectResponse;
import
 software.amazon.awssdk.services.mediastoredata.model.MediaStoreDataException;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.net.URI;
import java.net.URISyntaxException;

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

                Usage:    <completePath> <containerName> <savePath>

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

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

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

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

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

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

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

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

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

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

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

        DescribeContainerRequest containerRequest =
DescribeContainerRequest.builder()
```

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

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

- Untuk API detailnya, lihat GetObjectdi AWS SDK for Java 2.x APIReferensi.

Untuk daftar lengkap AWS SDKpanduan pengembang dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`ListContainers`** dengan AWS SDKatau CLI

Contoh kode berikut menunjukkan cara menggunakan`ListContainers`.

CLI

AWS CLI

Untuk melihat daftar kontainer

`list-containers`Contoh berikut menampilkan daftar semua kontainer yang terkait dengan akun Anda.

```
aws mediastore list-containers
```

Output:

```
{
    "Containers": [
        {
            "CreationTime": 1505317931,
            "Endpoint": "https://aaabbbcccdddee.data.mediastore.us-
west-2.amazonaws.com",
            "Status": "ACTIVE",
            "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleLiveDemo",
```

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

Untuk informasi selengkapnya, lihat Melihat Daftar Kontainer di AWS Panduan MediaStore Pengguna Elemental.

- Untuk API detailnya, lihat ListContainersdi AWS CLI Referensi Perintah.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di AWS Repositori Contoh Kode.

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

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
```

```
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListContainers {

    public static void main(String[] args) {

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

        listAllContainers(mediaStoreClient);
        mediaStoreClient.close();
    }

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

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

- Untuk API detailnya, lihat ListContainersdi AWS SDK for Java 2.x APIReferensi.

Untuk daftar lengkap AWS SDKpanduan pengembang dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`PutObject`** dengan AWS SDKatau CLI

Contoh kode berikut menunjukkan cara menggunakan`PutObject`.

CLI

AWS CLI

Untuk mengunggah objek

`put-object`Contoh berikut mengunggah objek ke wadah yang ditentukan. Anda dapat menentukan jalur folder tempat objek akan disimpan di dalam wadah. Jika folder sudah ada, AWS Elemental MediaStore menyimpan objek di folder. Jika folder tidak ada, layanan membuatnya, dan kemudian menyimpan objek di folder.

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

Output:

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

Untuk informasi selengkapnya, lihat Mengunggah Objek di AWS Panduan MediaStore Pengguna Elemental.

- Untuk API detailnya, lihat PutObjectdi AWS CLI Referensi Perintah.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di AWS Repositori Contoh Kode.

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

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

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

                Ex: <containerName> <filePath> <completePath>
```

```
                """;

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

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

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

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

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

            PutObjectRequest objectRequest = PutObjectRequest.builder()
                    .path(completePath)
                    .contentType("video/mp4")
                    .build();

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

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

```
    public static String getEndpoint(String containerName) {

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

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

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

- Untuk API detailnya, lihat PutObjectdi AWS SDK for Java 2.x APIReferensi.

Untuk daftar lengkap AWS SDKpanduan pengembang dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

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 DeleteObject	100	Jumlah maksimum permintaan operasi yang dapat Anda buat per detik. Permintaan tambahan diperlambat. Anda dapat meminta penambahan kuota.
Tingkat Permintaan API DescribeObject	1.000	Jumlah maksimum permintaan operasi yang dapat Anda buat per detik. Permintaan tambahan diperlambat. Anda dapat meminta penambahan kuota.
Tingkat permintaan GetObjectAPI untuk ketersediaan pengunggahan standar	1.000	Jumlah maksimum permintaan operasi yang dapat Anda buat per detik. Permintaan tambahan diperlambat. Anda dapat meminta penambahan kuota.
Tingkat permintaan GetObjectAPI untuk ketersediaan pengunggahan streaming	25	Jumlah maksimum permintaan operasi yang dapat Anda buat per detik. Permintaan tambahan diperlambat. Anda dapat meminta penambahan kuota.
Tingkat Permintaan API ListItems	5	Jumlah maksimum permintaan operasi yang dapat Anda buat per detik. Permintaan tambahan diperlambat. Anda dapat meminta penambahan kuota.

Sumber Daya atau Operasi	Kuota Default	Comments
Tingkat permintaan PutObjectAPI 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 meminta penambahan kuota. Dalam permintaan, tentukan TPS yang diminta dan ukuran objek rata-rata.
Tingkat permintaan PutObjectAPI untuk ketersediaan pengunggahan standar	100	Jumlah maksimum permintaan operasi yang dapat Anda buat per detik. Permintaan tambahan diperlambat. Anda dapat meminta penambahan kuota. 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 mempertajamAWS keterampilan Anda dan mendapatkan pengalaman praktis.
- AWSPusat Pengembang - Jelajahi tutorial, alat unduh, dan pelajari tentang acaraAWS pengembang.
- AWSAlat Pengembang — Tautan ke alat pengembang, SDK, kit alat IDE, dan alat baris perintah untuk mengembangkan dan mengelolaAWS aplikasi.
- Memulai Pusat Sumber Daya — Pelajari cara mengaturAkun AWS, bergabung denganAWS komunitas, dan meluncurkan aplikasi pertama Anda.
- Tutorial Hands-On - Ikuti step-by-step tutorial untuk meluncurkan aplikasi pertama AndaAWS.
- AWSLaporan Resmi — Tautan ke daftar lengkapAWS laporan resmi teknis, yang mencakup topik seperti arsitektur, keamanan, dan ekonomi dan ditulis oleh ArsitekAWS Solusi atau ahli teknis lainnya.
- AWS SupportCenter — Hub untuk membuat dan mengelolaAWS 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 tentangAWS Support, saluran dukungan respons cepat 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.
- AWSPersyaratan 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
Peningkatan Kontrol Akses Asal (OAC)	Menambahkan informasi tentang cara menggunakan OAC dengan AWS Elemental MediaStore.	17 April
Pembaruan kuota	Dikoreksi nilai kuota dan deskripsi untuk`Rules in a Metric Policy`.	25 Oktober
ExpiresAt bidang	Log akses sekarang menyertakan`ExpiresAt` bidang yang menunjukkan tanggal dan waktu kedaluwarsa objek berdasarkan aturan data sementara dalam kebijakan siklus hidup kontainer.	16 Juli
Aturan transisi siklus hidup	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

Wadah kosong	Anda sekarang dapat menghapus semua objek dalam wadah sekaligus.	7 April 2020
Support untuk CloudWatch metrik Amazon	Anda dapat menetapkan kebijakan metrik untuk menentukan metrik mana yang MediaStore dikirim CloudWatch.	30 Maret 2020
Wildcard dalam aturan objek hapus	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
Kebijakan siklus hidup objek	Sekarang Anda dapat menambahkan aturan ke kebijakan siklus hidup objek yang menunjukkan kedaluwarsa berdasarkan usia dalam hitungan detik.	13 September

AWS CloudFormationdukungan	Anda sekarang dapat menggunakanAWS CloudFormation template untuk membuat wadah secara otomatis. AWS CloudFormationTemplat 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

Kebijakan siklus hidup objek	Menambahkan dukungan untuk kebijakan siklus hidup objek, yang mengatur tanggal kedaluwarsa objek dalam wadah saat ini.	12 Desember 2018
Peningkatan kuota ukuran objek	Kuota untuk ukuran objek sekarang 25 MB.	10 Oktober 2018
Peningkatan kuota ukuran objek	Kuota untuk ukuran objek sekarang 20 MB.	6 September 2018
AWS CloudTrailintegrasi	Konten CloudTrail integrasi telah diperbarui agar selaras dengan perubahan terbaru pada CloudTrail layanan.	12 Juli 2018
Kolaborasi CDN	Menambahkan informasi tentang cara menggunakan AWS Elemental MediaStore dengan jaringan pengiriman konten (CDN) seperti Amazon CloudFront.	14 April
Konfigurasi CORS	AWS Elemental MediaStor e sekarang mendukung Cross-origin resource sharing (CORS), yang memungkin kan 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 penyimpan an video, AWS Elemental MediaStore, dan Panduan MediaStore Pengguna AWS Elemental.	27 November 2017

> **Note**
>
> - LayananAWS 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.