aws

Guida per l'utente

AWS Elemental MediaStore

AWS Elemental MediaStore: Guida per l’utente

Table of Contents

Avviso di fine del supporto: il 13 novembre 2025 AWS interromperà il supporto per AWS Elemental. MediaStore Dopo il 13 novembre 2025, non potrai più accedere alla console o alle MediaStore risorse. MediaStore Per ulteriori informazioni, consulta questo post del blog.

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.

Cos'è AWS Elemental MediaStore?

AWS Elemental MediaStore è un servizio di origine e archiviazione video che offre le alte prestazioni e la coerenza immediata necessarie per l'origine in tempo reale. Con MediaStore, puoi gestire le risorse video come oggetti in contenitori per creare flussi di lavoro multimediali affidabili e basati sul cloud.

Per usare il servizio, è possibile caricare gli oggetti da una sorgente, ad esempio un codificatore o feed di dati, in un container creato in MediaStore.

MediaStore è un'ottima scelta per archiviare file video frammentati quando è necessaria una forte coerenza, letture e scritture a bassa latenza e la capacità di gestire elevati volumi di richieste simultanee. Se non offri video in streaming live, prendi in considerazione l'utilizzo di Amazon Simple Storage Service (Amazon S3).

Argomenti

MediaStore Concetti e terminologia di AWS Elemental

ARN

Un Amazon Resource Name.

Body

I dati da caricare in un oggetto.

Intervallo (Byte)

Un sottoinsieme di dati di oggetto da esaminare. Per ulteriori informazioni, consulta intervallo dalla specifica HTTP.

Container

Uno spazio dei nomi che contiene gli oggetti. Un container ha un endpoint che è possibile utilizzare per scrivere e recuperare oggetti e collegare policy di accesso.

Endpoint

Un punto di accesso al MediaStore servizio, fornito come URL root HTTPS.

ETag

Un tag di entità che è un hash dei dati di oggetto.

Cartella

Una divisione di un container. Una cartella può contenere oggetti e altre cartelle.

Elemento

Termine utilizzato per fare riferimento a oggetti e cartelle.

Oggetto

Una risorsa, simile a un oggetto Amazon S3. Gli oggetti sono le entità fondamentali archiviate in MediaStore. Il servizio accetta tutti i tipi di file.

Servizio di emissione

MediaStore è considerato un servizio di origine perché è il punto di distribuzione per la distribuzione di contenuti multimediali.

Path

Un identificatore univoco di un oggetto o di una cartella, che ne indica la posizione nel container.

Parte

Un sottoinsieme di dati (blocco) di un oggetto.

Policy

Una policy IAM.

Risorsa

Un'entità in AWS che è possibile utilizzare. A ogni risorsa AWS viene assegnato un Amazon Resource Name (ARN) che agisce come un identificatore unico. In MediaStore, questa è la risorsa e il suo formato ARN:

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

Servizi correlati

- Amazon CloudFront è un servizio di rete di distribuzione di contenuti (CDN) globale che fornisce dati e video in modo sicuro ai tuoi spettatori. Puoi usare CloudFront per distribuire contenuti con le migliori prestazioni possibili. Per ulteriori informazioni, consulta l'Amazon CloudFront Developer Guide.
- CloudFormationè un servizio che ti aiuta a modellare e configurare AWS le tue risorse. Crei un modello che descrive tutte le AWS risorse che desideri (come i MediaStore contenitori) e CloudFormation si occupa del provisioning e della configurazione di tali risorse per te. Non è necessario creare e configurare singolarmente AWS le risorse e capire cosa dipende da cosa; CloudFormation gestisce tutto questo. Per ulteriori informazioni, consulta la AWS CloudFormation Guida per l'utente di .
- AWS CloudTrailè un servizio che ti consente di monitorare le chiamate effettuate all' CloudTrail API per il tuo account, incluse le chiamate effettuate dalla Console di gestione AWS e altri servizi. AWS CLI Per ulteriori informazioni, consulta la AWS CloudTrail Guida per l'utente di .
- Amazon CloudWatch è un servizio di monitoraggio per le risorse AWS cloud e le applicazioni su cui esegui AWS. Usa CloudWatch Events per tenere traccia delle modifiche allo stato dei contenitori e degli oggetti in MediaStore. Per ulteriori informazioni, consulta la CloudWatch documentazione di Amazon.
- AWS Identity and Access Management (IAM) è un servizio web che ti aiuta a controllare in modo sicuro l'accesso alle AWS risorse per i tuoi utenti. Usa IAM per controllare chi può utilizzare AWS le tue risorse (autenticazione) e quali risorse gli utenti possono utilizzare in quali modi (autorizzazione). Per ulteriori informazioni, consulta Configurazione di AWS Elemental MediaStore.
- Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) è uno storage di oggetti progettato per archiviare e recuperare qualsiasi quantità di dati da qualsiasi luogo. Per ulteriori informazioni, consulta la Documentazione di Amazon S3.

Accesso ad AWS Elemental MediaStore

È possibile accedere MediaStore utilizzando uno dei seguenti metodi:

- Console di gestione AWS: le procedure riportate in questa guida spiegano come utilizzare la Console di gestione AWS per eseguire attività per MediaStore. Per accedere MediaStore tramite la console:

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

- AWS Command Line Interface— Per ulteriori informazioni, consulta la Guida AWS Command Line Interface per l'utente. Per accedere MediaStore utilizzando l'endpoint CLI:

```
aws mediastore
```

- MediaStore API: se utilizzi un linguaggio di programmazione per il quale non è disponibile un SDK, consulta l'AWS Elemental MediaStore API Reference per informazioni sulle azioni API e su come effettuare richieste API. Per accedere MediaStore utilizzando l'endpoint dell'API REST:

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

- AWS SDKs: se utilizzi un linguaggio di programmazione per il quale AWS fornisce un SDK, puoi utilizzare un SDK per accedere. MediaStore SDKs semplifica l'autenticazione, si integra facilmente con il tuo ambiente di sviluppo e fornisci un facile accesso ai MediaStore comandi. Per ulteriori informazioni, consulta Strumenti per Amazon Web Services.
- AWS Tools per Windows PowerShell: per ulteriori informazioni, consulta la Guida per AWS Strumenti per PowerShell l'utente.

Prezzi per AWS Elemental MediaStore

Come per gli altri AWS prodotti, non sono previsti contratti o impegni minimi per l'utilizzo MediaStore. Verrà addebitata una tariffa di consumo per GB quando i contenuti arrivano al servizio e una tariffa mensile per GB per i contenuti archiviati nel servizio. Per ulteriori informazioni, consulta i prezzi di AWS Elemental MediaStore .

Regioni ed endpoint per AWS Elemental MediaStore

Per ridurre la latenza dei dati nelle tue applicazioni, MediaStore offre un endpoint regionale per effettuare la tua richiesta:

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

Per visualizzare l'elenco completo delle regioni AWS in cui MediaStore è disponibile, consulta gli MediaStore endpoint e le quote di AWS Elemental nell'AWS General Reference.

Configurazione di AWS Elemental MediaStore

Questa sezione ti guida attraverso i passaggi necessari per configurare gli utenti per accedere ad AWS MediaStore Elemental. Per informazioni di base e aggiuntive sulla gestione delle identità e degli accessi per MediaStore, consultaIdentity and Access Management per AWS Elemental MediaStore.

Per iniziare a usare AWS Elemental MediaStore, completa i seguenti passaggi.

Argomenti

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

   Parte della procedura di registrazione prevede la ricezione di una telefonata o di un messaggio di testo e l'immissione di un codice di verifica sulla tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a https://aws.amazon.com/e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi Console di gestione AWScome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

   Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .
2. In IAM Identity Center, assegna l'accesso amministrativo a un utente.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

  Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni conforme alla best practice dell'applicazione di autorizzazioni con il privilegio minimo.

   Segui le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente di AWS IAM Identity Center .

Guida introduttiva ad AWS Elemental MediaStore

Questo tutorial introduttivo mostra come usare AWS MediaStore Elemental per creare un contenitore e caricare un oggetto.

Argomenti

Fase 1: Accedi ad AWS Elemental MediaStore

Dopo aver configurato il tuo account AWS e creato utenti e ruoli, accedi alla console di AWS MediaStore Elemental.

Per accedere ad AWS Elemental MediaStore

- Accedi a Console di gestione AWS e apri la MediaStore console all'indirizzo https://console.aws.amazon.com/mediastore/.

> **Note**
>
> Puoi effettuare l'accesso utilizzando le credenziali IAM create per questo account. Per informazioni su come creare le credenziali IAM, consulta Configurazione di AWS Elemental MediaStore.

Fase 2: creare un container

Utilizzi contenitori in AWS MediaStore Elemental per archiviare cartelle e oggetti. I container consentono di raggruppare oggetti correlati nello stesso modo in cui si utilizza una directory per raggruppare i file in un file system. Non ti verrà addebitato alcun costo durante la creazione dei container; ti verranno addebitati i costi solo quando caricherai un oggetto in un container.

Per creare un container

1. Nella pagina Containers (Container), scegliere Create container (Crea container).
2. In Container name (Nome container) digita un nome per il container. Per ulteriori informazioni, consulta Regole per i nomi di container.
3. Scegli Crea contenitore. AWS Elemental MediaStore aggiunge il nuovo contenitore a un elenco di contenitori. Inizialmente, lo stato del container è Creating (In fase di creazione), quindi diventa Active (Attivo).

Fase 3: caricare un oggetto

Puoi caricare gli oggetti (con dimensioni massime di 25 MB per oggetto) in un container o in una cartella all'interno di un container. Per caricare un oggetto in una cartella, devi specificare il percorso della cartella. Se la cartella esiste già, AWS Elemental MediaStore memorizza l'oggetto nella cartella. Se la cartella non esiste, il servizio la crea e quindi archivia l'oggetto nella cartella.

> **Note**
>
> I nomi di file di oggetti possono contenere solo lettere, numeri, punti (.), trattini bassi (_), tilde (~) e trattini (-).

Per caricare un oggetto

1. Nella pagina Containers scegli il nome del container appena creato. Viene visualizzata la pagina dei dettagli del container.
2. Scegli Upload object (Carica oggetto).
3. In Target path (Percorso di destinazione) digita un percorso per le cartelle. Ad esempio, `premium/canada`. Se una delle cartelle nel percorso non esiste ancora, AWS Elemental la MediaStore crea automaticamente.
4. Per Object (Oggetto), scegli Browse (Sfoglia).
5. Passa alla cartella appropriata e scegli un oggetto da caricare.
6. Seleziona Open (Apri), quindi Upload (Carica).

Fase 4: accedere a un oggetto

Puoi scaricare i tuoi oggetti in un endpoint specificato.

1. Nella pagina Containers (Container), scegli il nome del container che contiene l'oggetto da scaricare.
2. Se l'oggetto che desideri scaricare si trova in una sottocartella, continua a selezionare i nomi di cartella fino a visualizzare l'oggetto.
3. Scegli il nome dell'oggetto.
4. Nella pagina dei dettagli per l'oggetto, scegli Download (Scarica).

Contenitori in AWS Elemental MediaStore

Utilizzi i contenitori MediaStore per archiviare cartelle e oggetti. Gli oggetti correlati possono essere raggruppati in container come si fa con una directory per raggruppare i file in un file system. Non ti verrà addebitato alcun costo durante la creazione dei container; ti verranno addebitati i costi solo quando caricherai un oggetto in un container. Per ulteriori informazioni sui costi, consulta la pagina dei prezzi di AWS Elemental MediaStore.

Argomenti

Regole per i nomi di container

Quando scegli un nome per il container, ricorda quanto segue:

- Il nome deve essere univoco all'interno dell'account corrente per la regione AWS corrente.
- Il nome può contenere lettere maiuscole e minuscole, numeri e caratteri di sottolineatura (_).
- Il nome deve contenere da 1 a 255 caratteri.
- I nomi rispettano la distinzione tra lettere maiuscole e minuscole. Ad esempio, puoi avere un container denominato `myContainer` e una cartella denominata `mycontainer` perché tali nomi sono univoci.
- Un container non può essere rinominato dopo che è stato creato.

Creazione di un container

Puoi creare fino a 100 container per ogni account AWS. Puoi creare il numero di cartelle che desideri, per non più di 10 livelli all'interno di un container. Inoltre, è possibile caricare il numero di oggetti che desideri in ogni contenitore.

> **Tip**
>
> Puoi anche creare un contenitore automaticamente utilizzando un CloudFormation modello. Il modello CloudFormation gestisce i dati per cinque operazioni API: creazione di un container, impostazione della registrazione degli accessi, aggiornamento della policy del container di default, aggiunta di una policy CORS e aggiunta della policy del ciclo di vita degli oggetti. Per ulteriori informazioni, consulta la AWS CloudFormation Guida per l'utente di .

Per creare un container (console)

1. Apri la MediaStore console all'indirizzo https://console.aws.amazon.com/mediastore/.
2. Nella pagina Containers (Container), scegliere Create container (Crea container).
3. In Container name (Nome container) immettere un nome per il container. Per ulteriori informazioni, consulta Regole per i nomi di container.
4. Scegli Crea contenitore. AWS Elemental MediaStore aggiunge il nuovo contenitore a un elenco di contenitori. Inizialmente, lo stato del container è Creating (In fase di creazione), quindi diventa Active (Attivo).

Per creare un container (AWS CLI)

- Nel AWS CLI, usa il `create-container` comando:

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

  L'esempio seguente mostra il valore restituito:

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

Visualizzazione dei dettagli di un container

I dettagli per un container includono la policy, l'endpoint, l'ARN e l'ora di creazione.

Per visualizzare i dettagli di un container (console)

1. Apri la console all' MediaStore indirizzo. https://console.aws.amazon.com/mediastore/
2. Nella pagina Containers (Container) scegliere il nome del container.

   Viene visualizzata la pagina dei dettagli del container. Questa pagina si articola in due sezioni:

   - La sezione Objects (Oggetti), in cui sono elencati gli oggetti e le cartelle nel container.
   - La sezione Container policy (Policy di container), che mostra la policy basata su risorse associata a questo container. Per ulteriori informazioni sulle policy basate su risorse, consultare Policy di container.

Per visualizzare i dettagli di un container (AWS CLI)

- In AWS CLI, usa il `describe-container` comando:

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

L'esempio seguente mostra il valore restituito:

```
{
    "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"
    }
}
```

Visualizzazione di un elenco di container

Puoi visualizzare un elenco di tutti i container associati al tuo account.

Per visualizzare un elenco di container (console)

- Apri la MediaStore console all'indirizzo https://console.aws.amazon.com/mediastore/.

  Viene visualizzata la pagina Containers (Container), con l'elenco di tutti i contenitori associati al tuo account.

Per visualizzare un elenco di container (AWS CLI)

- In AWS CLI, usa il `list-containers` comando.

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

  L'esempio seguente mostra il valore restituito:

  ```
  {
      "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"
          }
  ```

```
    ]
}
```

Eliminazione di un container

Puoi eliminare un container solo se non contiene oggetti.

Per eliminare un container (console)

1. Apri la MediaStore console all'indirizzo https://console.aws.amazon.com/mediastore/.
2. Nella pagina Containers (Container), scegliere l'opzione a sinistra del nome del container.
3. Scegliere Delete (Elimina).

Per eliminare un container (AWS CLI)

- In AWS CLI, usa il `delete-container` comando:

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

  Il comando non ha un valore restituito.

Policy in AWS Elemental MediaStore

Puoi applicare una o più di queste policy al tuo container AWS MediaStore Elemental:

- Policy del contenitore: imposta i diritti di accesso a tutte le cartelle e gli oggetti all'interno del contenitore. MediaStore imposta una politica predefinita che consente agli utenti di eseguire tutte MediaStore le operazioni sul contenitore. Questa policy specifica che tutte le operazioni devono essere eseguite su HTTPS. Dopo aver creato un container, puoi modificarne la policy.
- Politica CORS (Cross-Origin Resource Sharing): consente alle applicazioni Web client di un dominio di interagire con le risorse di un dominio diverso. MediaStore non imposta una politica CORS predefinita.
- Politica delle metriche: consente di MediaStore inviare metriche ad Amazon. CloudWatch MediaStore non imposta una politica metrica predefinita.
- Politica del ciclo di vita degli oggetti: controlla per quanto tempo gli oggetti rimangono in un contenitore. MediaStore MediaStorenon imposta una politica predefinita per il ciclo di vita degli oggetti.

Politiche relative ai container in AWS Elemental MediaStore

Ogni container presenta una policy basata su risorse che gestisce i diritti di accesso a tutte le cartelle e agli oggetti in tale container. La policy predefinita, che viene collegata automaticamente a tutti i nuovi contenitori, consente l'accesso a tutte le MediaStore operazioni AWS Elemental sul contenitore. e specifica che tale accesso ha la condizione di richiedere il protocollo HTTPS per le operazioni. Dopo aver creato un container, puoi modificare la policy collegata a tale container.

Puoi anche specificare una policy del ciclo di vita degli oggetti che regola la data di scadenza degli oggetti in un container. Dopo che gli oggetti raggiungono l'età massima specificata, il servizio elimina gli oggetti dal container.

Argomenti

- Visualizzazione di una policy di container
- Modifica di una policy di container
- Policy di container di esempio

Visualizzazione di una policy di container

Puoi utilizzare la console o visualizzare la politica basata AWS CLI sulle risorse di un contenitore.

Per visualizzare una policy di container (console)

1. Apri la console all' MediaStore indirizzo. https://console.aws.amazon.com/mediastore/
2. Nella pagina Containers (Container), scegliere il nome del container.

   Viene visualizzata la pagina dei dettagli del container. La policy viene visualizzata nella sezione Container policy (Policy container).

Per visualizzare una policy di container (AWS CLI)

- In AWS CLI, usa il `get-container-policy` comando:

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

L'esempio seguente mostra il valore restituito:

```
{
  "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"
          }
```

```
        }
      }
    ]
  }
}
```

Modifica di una policy di container

È possibile modificare le autorizzazioni nella policy di container predefinita o crearne una nuova per sostituirla. Affinché la nuova policy diventi effettiva, sono necessari fino a cinque minuti.

Per modificare una policy di container (console)

1. Apri la MediaStore console all'indirizzo https://console.aws.amazon.com/mediastore/.
2. Nella pagina Containers (Container), scegliere il nome del container.
3. Selezionare Edit policy (Modifica policy). Esempi di impostazione di autorizzazioni diverse sono disponibili su the section called "Policy di container di esempio".
4. Apportare le opportune modifiche e selezionare Save (Salva).

Per modificare una policy di container (AWS CLI)

1. Crea un file che definisca la policy del container:

   JSON

```
{
  "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. In AWS CLI, usa il `put-container-policy` comando:

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

Il comando non ha un valore restituito.

Policy di container di esempio

Gli esempi seguenti mostrano policy di container costruite per diversi gruppi di utenti.

Argomenti

- Policy di container di esempio: default
- Policy di container di esempio: accesso in lettura pubblico su HTTPS
- Policy di container di esempio: accesso in lettura pubblico su HTTP o HTTPS
- Policy di container di esempio: accesso in lettura multiaccount con abilitazione HTTP
- Policy di container di esempio: accesso in lettura multiaccount su HTTPS
- Policy di container di esempio: accesso in lettura multiaccount a un ruolo
- Policy di container di esempio: accesso completo multiaccount a un ruolo
- Policy di container di esempio: accesso limitato a indirizzi IP specifici

Policy di container di esempio: default

Quando crei un contenitore, AWS Elemental allega MediaStore automaticamente la seguente policy basata sulle risorse:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "MediaStoreFullAccess",
            "Action": [
                "mediastore:*"
            ],
            "Principal": {
                "AWS": "arn:aws:iam::333333333333:root"
            },
            "Effect": "Allow",
            "Resource": "arn:aws:mediastore:us-
east-2:333333333333:container/<container name>/*",
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "true"
                }
            }
        }
    ]
}
```

La policy è integrata nel servizio, quindi non è necessario crearla. Tuttavia, puoi modificare la policy sul contenitore se le autorizzazioni nella policy predefinita non sono in linea con le autorizzazioni che desideri utilizzare per il contenitore.

La policy predefinita assegnata a tutti i nuovi container consente l'accesso a tutte le operazioni di MediaStore sul container e specifica che tale accesso ha la condizione di richiedere il protocollo HTTPS per le operazioni.

Policy di container di esempio: accesso in lettura pubblico su HTTPS

Questa policy di esempio consente agli utenti di recuperare un oggetto tramite una richiesta HTTPS. Consente l'accesso in lettura a chiunque tramite una SSL/TLS connessione protetta: utenti autenticati e utenti anonimi (utenti che non hanno effettuato l'accesso). L'istruzione ha il nome `PublicReadOverHttps`. Consente l'accesso alle operazioni `GetObject` e `DescribeObject` su qualsiasi oggetto (come specificato dal simbolo * alla fine del percorso della risorsa) e specifica che tale accesso ha la condizione di richiedere il protocollo HTTPS per le operazioni:

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadOverHttps",
            "Effect": "Allow",
            "Action": [
                "mediastore:GetObject",
                "mediastore:DescribeObject"
            ],
            "Principal": "*",
            "Resource": "arn:aws:mediastore:us-
east-2:333333333333:container/<container name>/*",
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "true"
                }
            }
        }
    ]
}
```

Policy di container di esempio: accesso in lettura pubblico su HTTP o HTTPS

Questa policy di esempio consente l'accesso alle operazioni `GetObject` e `DescribeObject` su qualsiasi oggetto (come specificato dal simbolo * alla fine del percorso della risorsa). Consente accesso in lettura a chiunque, compresi tutti gli utenti autenticati e quelli anonimi (gli utenti che non sono connessi):

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadOverHttpOrHttps",
            "Effect": "Allow",
            "Action": [
                "mediastore:GetObject",
                "mediastore:DescribeObject"
            ],
            "Principal": "*",
```

```
            "Resource": "arn:aws:mediastore:us-
east-2:333333333333:container/<container name>/*",
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "false"
                }
            }
        }
    ]
}
```

Policy di container di esempio: accesso in lettura multiaccount con abilitazione HTTP

Questa policy di esempio consente agli utenti di recuperare un oggetto attraverso una richiesta HTTP. Consente l'accesso agli utenti autenticati con accesso multiaccount. Non è necessario che l'oggetto sia ospitato su un server con un certificato: SSL/TLS

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CrossAccountReadOverHttpOrHttps",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::333333333333:root"
            },
            "Action": [
                "mediastore:GetObject",
                "mediastore:DescribeObject"
            ],
            "Resource": "arn:aws:mediastore:us-
east-2:333333333333:container/<container name>/*",
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "true"
                }
            }
        }
    ]
```

```
}
```

Policy di container di esempio: accesso in lettura multiaccount su HTTPS

Questa politica di esempio consente l'accesso alle `DescribeObject` operazioni `GetObject` and su qualsiasi oggetto (come specificato dal simbolo* alla fine del percorso della risorsa) di proprietà dell'utente root dell'oggetto specificato<other acct number>. e specifica che tale accesso ha la condizione di richiedere il protocollo HTTPS per le operazioni:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CrossAccountReadOverHttps",
            "Effect": "Allow",
            "Action": [
                "mediastore:GetObject",
                "mediastore:DescribeObject"
            ],
            "Principal": {
                "AWS": "arn:aws:iam::333333333333:root"
            },
            "Resource": "arn:aws:mediastore:us-
east-2:333333333333:container/<container name>/*",
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "true"
                }
            }
        }
    ]
}
```

Policy di container di esempio: accesso in lettura multiaccount a un ruolo

La policy di esempio consente di accedere alle operazioni `GetObject` e `DescribeObject` su qualsiasi oggetto (come specificato dal simbolo * alla fine del percorso della risorsa) di proprietà

dell'account <numero account proprietario>. Consente l'accesso a qualsiasi utente dell'account <numero altro account> se tale account ha assunto il ruolo specificato in <nome ruolo>:

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

Policy di container di esempio: accesso completo multiaccount a un ruolo

Questa policy di esempio consente accesso multiaccount per aggiornare qualsiasi oggetto nell'account, se l'utente è connesso tramite HTTP. Inoltre, consente accesso multiaccount per eliminare, scaricare e descrivere gli oggetti su HTTP o HTTPS in un account che ha assunto il ruolo specificato:

- La prima istruzione è `CrossAccountRolePostOverHttps`. Consente l'accesso all'operazione `PutObject` su qualsiasi oggetto e consente l'accesso a qualsiasi utente dell'account specificato se tale account ha assunto il ruolo specificato in <nome ruolo>. Specifica che l'accesso ha la condizione di richiedere il protocollo HTTPS per l'operazione (tale condizione deve sempre essere inclusa quando si assegna l'accesso a `PutObject`).

  In altre parole, qualsiasi principale che abbia un accesso multiaccount può accedere a `PutObject`, ma solo tramite HTTPS.
- La seconda istruzione è `CrossAccountFullAccessExceptPost`. Consente l'accesso a tutte le operazioni tranne `PutObject` su qualsiasi oggetto. Consente questo accesso a qualsiasi utente dell'account specificato se tale account ha assunto il ruolo specificato in <nome ruolo>. Questo accesso non ha la condizione di richiedere il protocollo HTTPS per le operazioni.

In altre parole, qualsiasi account con accesso multiaccount può accedere a `DeleteObject`, `GetObject` e così via (ma non a `PutObject`) e può eseguire questa operazione su HTTP o HTTPS.

La seconda istruzione non sarà valida se non viene escluso `PutObject`, perché per includere `PutObject` è necessario impostare esplicitamente HTTPS come condizione.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CrossAccountRolePostOverHttps",
            "Effect": "Allow",
            "Action": "mediastore:PutObject",
            "Principal": {
                "AWS": "arn:aws:iam::333333333333:role/<role name>"
            },
            "Resource": "arn:aws:mediastore:us-
east-2:333333333333:container/<container name>/*",
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "true"
                }
            }
        },
        {
            "Sid": "CrossAccountFullAccessExceptPost",
            "Effect": "Allow",
            "NotAction": "mediastore:PutObject",
            "Principal": {
                "AWS": "arn:aws:iam::333333333333:role/<role name>"
            },
            "Resource": "arn:aws:mediastore:us-
east-2:333333333333:container/<container name>/*"
        }
    ]
}
```

Policy di container di esempio: accesso limitato a indirizzi IP specifici

Questa policy di esempio consente l'accesso a tutte le MediaStore operazioni AWS Elemental sugli oggetti nel contenitore specificato. La richiesta deve, tuttavia, avere origine dall'intervallo di indirizzi IP specificati nella condizione.

La condizione in questa dichiarazione identifica l'intervallo 198.51.100.* di indirizzi IP consentiti del protocollo Internet versione 4 (IPv4), con un'eccezione: 198.51.100.188.

Il blocco `Condition` utilizza le condizioni `IpAddress` e `NotIpAddress` e la chiave di condizione `aws:SourceIp`, che è una chiave di condizione AWS. `aws:sourceIp` IPv4 I valori utilizzano la notazione CIDR standard. Per ulteriori informazioni, consulta IP Address Condition Operators nella IAM User Guide.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AccessBySpecificIPAddress",
            "Effect": "Allow",
            "Action": [
                "mediastore:GetObject",
                "mediastore:DescribeObject"
            ],
            "Principal": "*",
            "Resource": "arn:aws:mediastore:us-
east-2:333333333333:container/<container name>/*",
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": [
                        "198.51.100.0/24"
                    ]
                },
                "NotIpAddress": {
                    "aws:SourceIp": "198.51.100.188/32"
                }
            }
        }
    ]
```

```
}
```

Policy di condivisione delle risorse tra origini (CORS) in AWS Elemental MediaStore

La funzionalità CORS (Cross-Origin Resource Sharing, condivisione delle risorse multiorigine) definisce un metodo con cui le applicazioni Web dei clienti caricate in un dominio possono interagire con le risorse situate in un dominio differente. Con il supporto CORS in AWS MediaStore Elemental, puoi creare ricche applicazioni Web lato client e consentire selettivamente l' MediaStore accesso multiorigine alle tue risorse. MediaStore

> **Note**
>
> Se utilizzi Amazon CloudFront per distribuire contenuti da un contenitore dotato di una policy CORS, assicurati di configurare la distribuzione per AWS MediaStore Elemental (inclusa la fase di modifica del comportamento della cache per configurare CORS).

In questa sezione viene fornita una panoramica della funzionalità CORS. I sottoargomenti descrivono come abilitare CORS utilizzando la console AWS MediaStore Elemental o a livello di codice utilizzando l'API MediaStore REST e AWS. SDKs

Argomenti

Scenari di casi d'uso di CORS

Di seguito sono riportati alcuni scenari di esempio per l'uso della funzionalità CORS.

- Scenario 1: supponiamo di distribuire video in streaming live in un contenitore AWS MediaStore Elemental denominato. LiveVideo I tuoi utenti caricano l'endpoint manifest del video `http://livevideo.mediastore.ap-southeast-2.amazonaws.com` da un'origine specifica come `www.example.com`. Vuoi utilizzare un lettore JavaScript video per accedere ai video che provengono da questo contenitore tramite richieste non autenticate. `GET` `PUT` Un browser in genere JavaScript impedisce di consentire tali richieste, ma puoi impostare una politica CORS sul tuo contenitore per abilitare esplicitamente queste richieste da. `www.example.com`
- Scenario 2: supponiamo di voler ospitare lo stesso live streaming dello Scenario 1 dal MediaStore contenitore, ma di voler consentire le richieste da qualsiasi origine. Puoi configurare una policy CORS per consentire origini contrassegnate con un carattere jolly (*), in modo che le richieste da qualsiasi origine possono accedere al video.

Aggiunta di una policy CORS a un container

Questa sezione spiega come aggiungere una configurazione CORS (Cross-Origin Resource Sharing) a un contenitore AWS MediaStore Elemental. La funzionalità CORS consente l'interazione tra le applicazioni client Web caricate in un dominio e le risorse situate in un altro dominio.

Per configurare il container per permettere richieste multiorigine, aggiungi una policy CORS al container. La policy CORS definisce le regole che identificano le origini che potranno accedere al container, le operazioni (metodi HTTP) supportate per ogni origine e altre informazioni specifiche dell'operazione.

Quando aggiungi una policy CORS al container, le policy del container (che disciplinano i diritti di accesso al container) continueranno a essere applicate.

Per aggiungere una policy CORS (console)

1. Apri la console all' MediaStore indirizzo. https://console.aws.amazon.com/mediastore/
2. Nella pagina Containers (Container), scegli il nome del container per il quale vuoi creare una policy CORS.

   Viene visualizzata la pagina dei dettagli del container.
3. Nella sezione Container CORS policy (Policy CORS del container) scegli Create CORS policy (Crea policy CORS).
4. Inserisci la policy in formato JSON e quindi scegli Save (Salva).

Per aggiungere una policy CORS (AWS CLI)

1. Creare un file che definisca la policy CORS:

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

2. In AWS CLI, usa il `put-cors-policy` comando.

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

   Il comando non ha un valore restituito.

Visualizzazione di una policy CORS

La funzionalità CORS (Cross-Origin Resource Sharing, condivisione delle risorse multiorigine) definisce un metodo con cui le applicazioni Web dei clienti caricate in un dominio possono interagire con le risorse situate in un dominio differente.

Per visualizzare una policy CORS (console)

1. Apri la MediaStore console all'indirizzo https://console.aws.amazon.com/mediastore/.
2. Nella pagina Containers (Container), scegli il nome del container di cui vuoi visualizzare la policy CORS.

Viene visualizzata la pagina dei dettagli del container con la policy CORS nella sezione Container CORS policy (Policy CORS del container).

Per visualizzare una policy CORS (AWS CLI)

- In AWS CLI, usa il `get-cors-policy` comando:

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

L'esempio seguente mostra il valore restituito:

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

Modifica di una policy CORS

La funzionalità CORS (Cross-Origin Resource Sharing, condivisione delle risorse multiorigine) definisce un metodo con cui le applicazioni Web dei clienti caricate in un dominio possono interagire con le risorse situate in un dominio differente.

Per modificare una policy CORS (console)

1. Apri la MediaStore console all'indirizzo https://console.aws.amazon.com/mediastore/.

2. Nella pagina Containers (Container), scegli il nome del container di cui vuoi modificare la policy CORS.

   Viene visualizzata la pagina dei dettagli del container.

3. Nella sezione Container CORS policy (Policy CORS del container) scegli Edit CORS policy (Modifica policy CORS).
4. Effettua le modifiche alla policy, quindi scegli Save (Salva).

Per modificare una policy CORS (AWS CLI)

1. Creare un file che definisca la policy CORS aggiornata:

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

2. In AWS CLI, usa il `put-cors-policy` comando.

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

   Il comando non ha un valore restituito.

Eliminazione di una policy CORS

La funzionalità CORS (Cross-Origin Resource Sharing, condivisione delle risorse multiorigine) definisce un metodo con cui le applicazioni Web dei clienti caricate in un dominio possono interagire

con le risorse situate in un dominio differente. L'eliminazione di una policy CORS da un container rimuove le autorizzazioni per le richieste multiorigine.

Per eliminare una policy CORS (console)

1. Apri la MediaStore console all'indirizzo https://console.aws.amazon.com/mediastore/.
2. Nella pagina Containers (Container), scegli il nome del container di cui vuoi eliminare la policy CORS.

   Viene visualizzata la pagina dei dettagli del container.
3. Nella sezione Container CORS policy (Policy CORS del container) scegli Delete CORS policy (Elimina policy CORS).
4. Scegli Continue (Continua) per confermare, quindi scegli Save (Salva).

Per eliminare una policy CORS (AWS CLI)

- In AWS CLI, usa il `delete-cors-policy` comando:

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

  Il comando non ha un valore restituito.

Risoluzione dei problemi correlati alla configurazione CORS

Se si verifica un comportamento imprevisto quando accedi a un container che dispone di una policy CORS, segui questa procedura per risolvere il problema.

1. Verifica che la policy CORS sia collegata al container.

   Per istruzioni, consulta the section called "Visualizzazione di una policy CORS".
2. Acquisisci la richiesta e la risposta complete utilizzando uno strumento di tua scelta (ad esempio la console di sviluppo del browser). Verifica che la policy CORS collegata al container includa almeno una regola CORS che soddisfi i dati nella richiesta, come segue:

   a. Verifica che la richiesta abbia un'intestazione `Origin`.

Se manca l'intestazione, AWS MediaStore Elemental non tratta la richiesta come una richiesta multiorigine e non invia le intestazioni di risposta CORS nella risposta.

b. Verifica che l'intestazione `Origin` nella richiesta corrisponda ad almeno uno degli elementi `AllowedOrigins` nella `CORSRule` specifica.

   Lo schema, l'host e i valori della porta nell'intestazione della richiesta `Origin` devono corrispondere a `AllowedOrigins` in `CORSRule`. Se ad esempio imposti `CORSRule` per consentire l'origine `http://www.example.com`, nessuna delle due origini `https://www.example.com` e `http://www.example.com:80` nella richiesta corrisponde all'origine consentita nella configurazione.

c. Verifica che il metodo nella richiesta (o il metodo specificato in `Access-Control-Request-Method` in caso di una richiesta preliminare) corrisponda a uno degli elementi `AllowedMethods` nella stessa `CORSRule`.

d. Per una richiesta preliminare, se la richiesta include un'intestazione `Access-Control-Request-Headers`, verificare che la `CORSRule` includa le voci `AllowedHeaders` per ogni valore nell'intestazione `Access-Control-Request-Headers`.

Policy CORS di esempio

I seguenti esempi mostrano le policy CORS (Cross-Origin Resource Sharing).

Argomenti

Policy CORS di esempio: accesso in lettura per qualsiasi dominio

La seguente policy consente a una pagina Web di qualsiasi dominio di recuperare contenuti dal tuo contenitore AWS MediaStore Elemental. La richiesta include tutte le intestazioni HTTP dal dominio di origine e il servizio risponde solo alle richieste HTTP GET e HTTP HEAD dal dominio di origine. I risultati vengono memorizzati nella cache per 3.000 secondi prima della consegna di un nuovo set di risultati.

```
[
  {
    "AllowedHeaders": [
```

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

Policy CORS di esempio: accesso in lettura per un dominio specifico

La seguente policy consente a una pagina Web da `https://www.example.com` cui recuperare contenuti dal container AWS MediaStore Elemental. La richiesta include tutte le intestazioni HTTP da `https://www.example.com`e il servizio risponde solo alle richieste HTTP GET e HTTP HEAD da `https://www.example.com`. I risultati vengono memorizzati nella cache per 3.000 secondi prima della consegna di un nuovo set di risultati.

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

Politiche del ciclo di vita degli oggetti in AWS Elemental MediaStore

Per ogni container, puoi creare una policy del ciclo di vita degli oggetti che gestisce la durata di archiviazione degli oggetti nel container. Quando gli oggetti raggiungono l'età massima specificata,

AWS Elemental li MediaStore elimina. Puoi eliminare gli oggetti quando non sono più necessari per risparmiare sui costi di storage.

Puoi anche specificare che gli oggetti MediaStore devono essere spostati nella classe di storage ad accesso infrequente (IA) dopo aver raggiunto una certa età. Gli oggetti archiviati nella classe di archiviazione IA hanno velocità diverse per l'archiviazione e il recupero rispetto agli oggetti archiviati nella classe di archiviazione standard. Per ulteriori informazioni, consulta MediaStore Prezzi.

Una policy del ciclo di vita degli oggetti contiene le regole che definiscono la durata di oggetti per sottocartella. Non puoi assegnare una policy del ciclo di vita degli oggetti a singoli oggetti. Puoi collegare una sola policy del ciclo di vita degli oggetti a un container, ma puoi aggiungere fino a 10 regole a ogni policy del ciclo di vita degli oggetti. Per ulteriori informazioni, consulta Componenti di una policy del ciclo di vita degli oggetti.

Argomenti

Componenti di una policy del ciclo di vita degli oggetti

Le policy del ciclo di vita degli oggetti regolano per quanto tempo gli oggetti rimangono in un contenitore AWS Elemental. MediaStore Ogni policy del ciclo di vita degli oggetti è costituita da una o più regole, che determinano la durata degli oggetti. Una regola può essere associata a una cartella, più cartelle o l'intero container.

Puoi collegare una policy del ciclo di vita degli oggetti a un container e ogni policy del ciclo di vita degli oggetti può contenere fino a 10 regole. Non puoi assegnare una policy del ciclo di vita degli oggetti a un singolo oggetto.

Regole in una policy del ciclo di vita degli oggetti

È possibile creare tre tipi di regole:

- Dati transitori
- Eliminazione dell'oggetto
- Transizione del ciclo di vita

Dati transitori

Una regola di dati transitoria imposta la scadenza degli oggetti entro pochi secondi. Questo tipo di regola si applica solo agli oggetti aggiunti al container dopo che la policy diventa efficace. Sono necessari fino a 20 minuti per MediaStore applicare la nuova policy al contenitore.

Un esempio di regola per i dati transitori è simile alla seguente:

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

Le regole dei dati transitori hanno tre parti:

- `path`: sempre impostato su `wildcard`. Utilizza questa parte per definire gli oggetti da eliminare. Puoi utilizzare uno o più caratteri jolly, rappresentati da un asterisco (*). Ogni carattere jolly rappresenta qualsiasi combinazione di zero o più caratteri. Ad esempio, `"path": [ {"wildcard": "Football/index*.m3u8"} ],` si applica a tutti i file nella cartella `Football` che corrispondono al modello di `index*.m3u8` (ad esempio index.m3u8, index1.m3us8 e index123456.m3u8). Puoi includere fino a 10 percorsi in un'unica regola.
- `seconds_since_create`: sempre impostato su `numeric`. Puoi specificare un valore compreso tra 1 e 300 secondi Puoi anche impostare l'operatore su maggiore di (**>**) oppure maggiore o uguale a (**>=**).
- `action`: sempre impostato su `EXPIRE`.

Per le regole di dati transitori (gli oggetti scadono in pochi secondi), non vi è alcun ritardo tra la scadenza di un oggetto e l'eliminazione dell'oggetto.

> **Note**
>
> Gli oggetti soggetti a una regola di dati transitori non sono inclusi nella risposta di `list-items`. Inoltre, gli oggetti che scadono a causa di una regola di dati transitoria non emettono alcun CloudWatch evento alla scadenza.

Eliminazione dell'oggetto

Una regola di eliminazione dell'oggetto imposta la scadenza degli oggetti entro pochi giorni. Questo tipo di regola si applica a tutti gli oggetti nel container, anche se sono stati aggiunti al container prima della creazione della policy. Sono necessari fino a 20 minuti per MediaStore applicare la nuova policy, ma possono essere necessarie fino a 24 ore prima che gli oggetti vengano rimossi dal contenitore.

Un esempio di due regole per l'eliminazione di oggetti è simile al seguente:

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

Le regole dell'oggetto di eliminazione hanno tre parti:

- `path`: impostare su `prefix` o su `wildcard`. Non puoi mescolare `prefix` e `wildcard` nella stessa regola. Se desideri utilizzare entrambi, è necessario creare una regola per `prefix` e una regola separata per `wildcard`, come mostrato nell'esempio precedente.

  - `prefix` – Puoi impostare il percorso su `prefix` se desideri eliminare tutti gli oggetti all'interno di una determinata cartella. Se il parametro è vuoto (`"path": [ { "prefix": "" } ],`), la destinazione è tutti gli oggetti archiviati ovunque all'interno del container corrente. Puoi includere fino a 10 percorsi `prefix` in un'unica regola.
  - `wildcard` – Per eliminare oggetti specifici in base al nome del file e/o al tipo di file imposti il percorso su `wildcard`. Puoi utilizzare uno o più caratteri jolly, rappresentati da un asterisco (*). Ogni carattere jolly rappresenta qualsiasi combinazione di zero o più caratteri. Ad esempio, `"path": [ {"wildcard": "Football/*.ts"} ],` si applica a tutti i file della cartella `Football` che corrispondono al modello di `*.ts` (ad esempio filename.ts, filename1.ts e filename123456.ts). Puoi includere fino a 10 percorsi `wildcard` in un'unica regola.
- `days_since_create`: sempre impostato su `numeric`. Puoi specificare un valore compreso tra 1 e 36.500 giorni. Puoi anche impostare l'operatore su maggiore di (**>**) oppure maggiore o uguale a (**>=**).
- `action`: sempre impostato su `EXPIRE`.

Per le regole di eliminazione degli oggetti (gli oggetti scadono entro pochi giorni), potrebbe esserci un leggero ritardo tra la scadenza di un oggetto e l'eliminazione dell'oggetto. Tuttavia, le modifiche nella fatturazione avvengono non appena l'oggetto scade. Ad esempio, se una regola del ciclo di vita specifica 10 `days_since_create`, l'account non viene fatturato per l'oggetto dopo 10 giorni, anche se l'oggetto non è ancora stato eliminato.

Transizione del ciclo di vita

Una regola di transizione del ciclo di vita imposta gli oggetti da spostare nella classe di archiviazione con accesso non frequente (IA) dopo aver raggiunto una certa età, misurata in giorni. Gli oggetti archiviati nella classe di archiviazione IA hanno velocità diverse per l'archiviazione e il recupero rispetto agli oggetti archiviati nella classe di archiviazione standard. Per ulteriori informazioni, consulta la sezione Prezzi di MediaStore.

Una volta che un oggetto è stato spostato nella classe di archiviazione IA, non è possibile riportarlo alla classe di archiviazione standard.

La regola di transizione del ciclo di vita si applica a tutti gli oggetti nel container, anche se sono stati aggiunti al container prima della creazione della policy. Sono necessari fino a 20 minuti per MediaStore applicare la nuova policy, ma possono essere necessarie fino a 24 ore prima che gli oggetti vengano rimossi dal contenitore.

Un esempio di una regola di transizione del ciclo di vita è simile a questo:

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

Le regole di transizione del ciclo di vita hanno tre parti:

- `path`: impostare su `prefix` o su `wildcard`. Non puoi mescolare `prefix` e `wildcard` nella stessa regola. Se desideri utilizzare entrambi, devi creare una regola per `prefix` e una regola separata per `wildcard`.
  - `prefix` - Imposti il percorso su `prefix` se desideri passare tutti gli oggetti all'interno di una particolare cartella alla classe di archiviazione IA. Se il parametro è vuoto (`"path": [ { "prefix": "" } ],`), la destinazione è tutti gli oggetti archiviati ovunque all'interno del container corrente. Puoi includere fino a 10 percorsi `prefix` in un'unica regola.
  - `wildcard` - Imposti il percorso su `wildcard` se desideri passare oggetti specifici alla classe di archiviazione IA in base al nome del file e/o al tipo di file. Puoi utilizzare uno o più caratteri jolly, rappresentati da un asterisco (*). Ogni carattere jolly rappresenta qualsiasi combinazione di zero o più caratteri. Ad esempio, `"path": [ {"wildcard": "Football/*.ts"} ],` si applica a tutti i file della cartella `Football` che corrispondono al modello di `*.ts` (ad esempio filename.ts, filename1.ts e filename123456.ts). Puoi includere fino a 10 percorsi `wildcard` in un'unica regola.
- `days_since_create`: sempre impostato su `"numeric": [">=" , 30]`.
- `action`: sempre impostato su `ARCHIVE`.

Esempio

Ad esempio, un container denominato `LiveEvents` dispone di quattro sottocartelle: `Football`, `Baseball`, `Basketball` e `AwardsShow`. L'aspetto della policy del ciclo di vita degli oggetti assegnata alla cartella `LiveEvents` è simile al seguente:

```
{
    "rules": [
         {
            "definition": {
                "path": [
                    {"prefix": "Football/"},
                    {"prefix": "Baseball/"}
                ],
                "days_since_create": [
                    {"numeric": [">" , 28]}
                ]
            },
            "action": "EXPIRE"
        },
        {
            "definition": {
                "path": [ { "prefix": "AwardsShow/" }  ],
                "days_since_create": [
                    {"numeric": [">=" , 15]}
                ]
            },
            "action": "EXPIRE"
        },
        {
            "definition": {
                "path": [ { "prefix": "" }  ],
                "days_since_create": [
                    {"numeric": [">" , 40]}
                ]
            },
            "action": "EXPIRE"
        },
        {
            "definition": {
                "path": [ { "wildcard": "Football/*.ts" }  ],
                "days_since_create": [
                    {"numeric": [">" , 20]}
                ]
            },
            "action": "EXPIRE"
        },
        {
            "definition": {
```

```
                "path": [
                    {"wildcard": "Football/index*.m3u8"}
                ],
                "seconds_since_create": [
                    {"numeric": [">" , 15]}
                ]
            },
            "action": "EXPIRE"
        },
        {
                "definition": {
                    "path": [
                        {"prefix": "Program/"}
                    ],
                    "days_since_create": [
                        {"numeric": [">=" , 30]}
                    ]
                },
                "action": "ARCHIVE"
            }
    ]
}
```

La policy precedente specifica quanto segue:

- La prima regola indica ad AWS MediaStore Elemental di eliminare gli oggetti archiviati nella cartella e `LiveEvents/Football` `LiveEvents/Baseball` nella cartella dopo che sono più vecchi di 28 giorni.
- La seconda regola impone al servizio di eliminare gli oggetti archiviati nella cartella `LiveEvents/AwardsShow` quando sono più vecchi di 15 giorni.
- La terza regola impone al servizio di eliminare gli oggetti archiviati in qualsiasi parte del container `LiveEvents` quando sono più vecchi di 40 giorni. Questa regola si applica a oggetti archiviati direttamente nel container `LiveEvents`, nonché a oggetti archiviati in una qualsiasi delle quattro sottocartelle del container.
- La quarta regola indica al servizio di eliminare gli oggetti nella cartella `Football` che corrispondono al modello `*.ts` quando sono più vecchi di 20 giorni.
- La quinta regola indica al servizio di eliminare gli oggetti nella `Football` cartella che corrispondono allo schema `index*.m3u8` dopo che sono più vecchi di 15 secondi. MediaStore elimina questi file 16 secondi dopo che sono stati inseriti nel contenitore.

- La sesta regola indica al servizio di spostare gli oggetti nella cartella `Program` nella classe di archiviazione IA dopo 30 giorni.

Per altri esempi di policy relative al ciclo di vita degli oggetti, consulta Esempio di policy del ciclo di vita degli oggetti.

Aggiunta di una policy del ciclo di vita degli oggetti a un container

Una policy del ciclo di vita degli oggetti consente di specificare la durata di archiviazione degli oggetti in un container. Imposta una data di scadenza e dopo la data di scadenza AWS Elemental MediaStore elimina gli oggetti. Sono necessari fino a 20 minuti affinché il servizio applichi la nuova policy al container.

Per informazioni su come creare una policy del ciclo di vita, consulta Componenti di una policy del ciclo di vita degli oggetti.

> **Note**
>
> Per le regole di eliminazione degli oggetti (gli oggetti scadono entro pochi giorni), potrebbe esserci un leggero ritardo tra la scadenza di un oggetto e l'eliminazione dell'oggetto. Tuttavia, le modifiche nella fatturazione avvengono non appena l'oggetto scade. Ad esempio, se una regola del ciclo di vita specifica 10 `days_since_create`, l'account non viene fatturato per l'oggetto dopo 10 giorni, anche se l'oggetto non è ancora stato eliminato.

Per aggiungere una policy del ciclo di vita degli oggetti (console)

1. Apri la MediaStore console all'indirizzo. https://console.aws.amazon.com/mediastore/
2. Nella pagina Containers (Container), scegli il nome del container per il quale vuoi creare una policy del ciclo di vita degli oggetti.

   Viene visualizzata la pagina dei dettagli del container.
3. Nella sezione Object lifecycle policy (Policy del ciclo di vita degli oggetti), scegliere Create object lifecycle policy (Crea policy del ciclo di vita degli oggetti).
4. Inserisci la policy in formato JSON e quindi scegli Save (Salva).

Per aggiungere una policy del ciclo di vita degli oggetti (AWS CLI)

1. Crea un file che definisce la policy del ciclo di vita degli oggetti:

```
{
    "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. In AWS CLI, usa il `put-lifecycle-policy` comando:

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

   Il comando non ha un valore restituito. Il servizio collega la policy specificata al container.

Visualizzazione di una policy del ciclo di vita degli oggetti

Una policy del ciclo di vita degli oggetti specifica per quanto tempo gli oggetti devono essere conservati in un container.

Per visualizzare una policy del ciclo di vita di un oggetto (console)

1. Apri la MediaStore console all'indirizzo https://console.aws.amazon.com/mediastore/.
2. Nella pagina Containers (Container), scegli il nome del container di cui vuoi visualizzare la policy del ciclo di vita degli oggetti.

   Viene visualizzata la pagina dei dettagli del container, con la policy del ciclo di vita degli oggetti nella sezione Object lifecycle policy (Policy del ciclo di vita degli oggetti).

Per visualizzare una policy del ciclo di vita degli oggetti (AWS CLI)

- In AWS CLI, usa il `get-lifecycle-policy` comando:

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

  L'esempio seguente mostra il valore restituito:

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

Modifica di una policy del ciclo di vita degli oggetti

Non puoi modificare una policy del ciclo di vita degli oggetti esistente. Tuttavia, puoi modificare una policy esistente caricando una policy di sostituzione. Sono necessari fino a 20 minuti affinché il servizio applichi la policy aggiornata al container.

Per modificare una policy del ciclo di vita degli oggetti (console)

1. Apri la MediaStore console all'indirizzo https://console.aws.amazon.com/mediastore/.
2. Nella pagina Containers (Container), scegli il nome del container di cui vuoi modificare la policy del ciclo di vita degli oggetti.

   Viene visualizzata la pagina dei dettagli del container.
3. Nella sezione Object lifecycle policy (Policy del ciclo di vita degli oggetti), scegliere Edit object lifecycle policy (Modifica policy del ciclo di vita degli oggetti).
4. Effettua le modifiche alla policy, quindi scegli Save (Salva).

Per modificare una policy del ciclo di vita degli oggetti (AWS CLI)

1. Crea un file che definisce la policy del ciclo di vita degli oggetti aggiornata:

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

2. In AWS CLI, usa il `put-lifecycle-policy` comando:

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

Il comando non ha un valore restituito. Il servizio collega la policy specificata al container, sostituendo la policy precedente.

Eliminazione di una policy del ciclo di vita degli oggetti

Quando elimini una policy del ciclo di vita dell'oggetto, sono necessari fino a 20 minuti affinché il servizio applichi la modifica al container.

Per eliminare una policy del ciclo di vita degli oggetti (console)

1. Apri la MediaStore console all'indirizzo https://console.aws.amazon.com/mediastore/.
2. Nella pagina Containers (Container), scegli il nome del container di cui vuoi eliminare la policy del ciclo di vita degli oggetti.

   Viene visualizzata la pagina dei dettagli del container.
3. Nella sezione Object lifecycle policy (Policy del ciclo di vita degli oggetti), scegliere Delete lifecycle policy (Elimina policy del ciclo di vita degli oggetti).
4. Scegli Continue (Continua) per confermare, quindi scegli Save (Salva).

Per eliminare una policy del ciclo di vita degli oggetti (AWS CLI)

- In AWS CLI, usa il `delete-lifecycle-policy` comando:

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

Il comando non ha un valore restituito.

Esempio di policy del ciclo di vita degli oggetti

Negli esempi seguenti vengono illustrate le policy relative al ciclo di vita degli oggetti.

Argomenti

- Policy di esempio relativa al ciclo di vita degli oggetti: scadenza in pochi secondi
- Policy di esempio relative al ciclo di vita degli oggetti: scadenza entro alcun giorni
- Policy di esempio relative al ciclo di vita degli oggetti: transizione alla classe di archiviazione con accesso non frequente
- Policy di esempio del ciclo di vita degli oggetti: regole multiple
- Policy di esempio del ciclo di vita degli oggetti: container vuoto

Policy di esempio relativa al ciclo di vita degli oggetti: scadenza in pochi secondi

La seguente politica specifica che MediaStore vengono eliminati gli oggetti che soddisfano tutti i seguenti criteri:

- L'oggetto è stato aggiunto al container dopo che la policy era divenuta efficace.
- L'oggetto è memorizzato nella cartella `Football`.
- L'oggetto ha un'estensione del file di `m3u8`.
- L'oggetto è stato nel container per più di 20 secondi.

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

Policy di esempio relative al ciclo di vita degli oggetti: scadenza entro alcun giorni

La seguente politica specifica che MediaStore vengono eliminati gli oggetti che soddisfano tutti i seguenti criteri:

- L'oggetto è memorizzato nella `Program` cartella
- L'oggetto ha un'estensione del file di `ts`
- L'oggetto è rimasto nel container per più di 5 giorni

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

Policy di esempio relative al ciclo di vita degli oggetti: transizione alla classe di archiviazione con accesso non frequente

La seguente politica specifica che gli oggetti vengono MediaStore spostati nella classe di archiviazione Infrequent Access (IA) quando hanno 30 giorni. Gli oggetti archiviati nella classe di archiviazione IA hanno velocità diverse per l'archiviazione e il recupero rispetto agli oggetti archiviati nella classe di archiviazione standard.

Il campo `days_since_create` deve essere impostato su `"numeric": [">=" ,30]`.

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

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

Policy di esempio del ciclo di vita degli oggetti: regole multiple

La seguente politica specifica che MediaStore esegue le seguenti operazioni:

- Spostare gli oggetti memorizzati nella cartella `AwardsShow` nella classe di archiviazione con accesso non frequente (IA) dopo 30 giorni
- Eliminare gli oggetti che hanno un'estensione del file di `m3u8` e che sono memorizzati nella cartella `Football` dopo 20 secondi
- Eliminare gli oggetti memorizzati nella cartella `April` dopo 10 giorni
- Eliminare gli oggetti che hanno un'estensione di file `ts` e che sono memorizzati nella cartella `Program` dopo 5 giorni

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

Policy di esempio del ciclo di vita degli oggetti: container vuoto

La seguente politica del ciclo di vita degli oggetti specifica che tutti gli oggetti nel contenitore, incluse cartelle e sottocartelle, vengono MediaStore eliminati 1 giorno dopo l'aggiunta al contenitore. Se il contenitore contiene oggetti prima dell'applicazione di questa politica, MediaStore elimina gli oggetti 1 giorno dopo l'entrata in vigore della politica. Sono necessari fino a 20 minuti affinché il servizio applichi la nuova policy al container.

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

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

Policy metriche in AWS Elemental MediaStore

Per ogni contenitore, puoi aggiungere una policy di metrica per consentire ad AWS MediaStore Elemental di inviare parametri ad Amazon. CloudWatch Affinché la nuova policy diventi effettiva, sono necessari fino a 20 minuti. Per una descrizione di ogni MediaStore metrica, consulta. MediaStore metriche

Un policy di parametro contiene quanto segue:

- Impostazione per abilitare o disabilitare i parametri a livello di container.
- Da zero a cinque regole che abilitano i parametri a livello di oggetto. Se la policy contiene regole, ogni regola deve includere entrambi i seguenti elementi:
  - Gruppo di oggetti che definisce gli oggetti da includere nel gruppo. La definizione può essere un percorso o un nome di file, ma non può contenere più di 900 caratteri. I caratteri validi sono: a-z, A-Z, 0-9, _ (carattere di sottolineatura), = (uguale), : (due punti), . (punto), - (trattino), ~ (tilde), / (barra) e * (asterisco). I caratteri jolly (*) sono accettabili.
  - Nome di un gruppo di oggetti che consente di fare riferimento al gruppo di oggetti. Il nome non può contenere più di 30 caratteri. I caratteri validi sono: a-z, A-Z, 0-9 e _ (carattere di sottolineatura).

Se un oggetto soddisfa più regole, CloudWatch visualizza un punto dati per ogni regola di corrispondenza. Ad esempio, se un oggetto corrisponde a due regole denominate `rule1` `erule2`, CloudWatch visualizza due punti dati per queste regole. Il primo ha la dimensione `ObjectGroupName=rule1`, mentre per il secondo la dimensione è `ObjectGroupName=rule2`.

Argomenti

- Aggiunta di una policy di parametro
- Visualizzazione di una policy di parametro

Aggiunta di una policy di parametro

Una policy sui parametri contiene regole che determinano quali metriche AWS Elemental invia ad Amazon. MediaStore CloudWatch Per esempi di policy di parametro, consulta Policy di parametro di esempio.

Per aggiungere una policy di parametro (console)

1. Apri la console all'indirizzo. MediaStore https://console.aws.amazon.com/mediastore/
2. Nella pagina Containers (Container), scegli il nome del container a cui aggiungere la policy di parametro.

   Viene visualizzata la pagina dei dettagli del container.
3. Nella sezione Metric policy (Policy di parametro), scegli Create metric policy (Crea policy di parametro).
4. Inserisci la policy in formato JSON e quindi scegli Save (Salva).

Visualizzazione di una policy di parametro

Puoi utilizzare la console o AWS CLI visualizzare la politica dei parametri di un contenitore.

Per visualizzare una policy di parametro (console)

1. Apri la MediaStore console all'indirizzo https://console.aws.amazon.com/mediastore/.
2. Nella pagina Containers (Container), scegliere il nome del container.

   Viene visualizzata la pagina dei dettagli del container. La policy viene visualizzata nella sezione Metric policy (Policy di parametro).

Modifica di una policy di parametro

Una policy sui parametri contiene regole che determinano quali metriche AWS Elemental invia ad Amazon. MediaStore CloudWatch Quando si modifica una policy di parametro esistente, occorrono

fino a 20 minuti prima che la nuova policy abbia effetto. Per esempi di policy di parametro, consulta Policy di parametro di esempio.

Per modificare una policy di parametro (console)

1. Apri la console all'indirizzo. MediaStore https://console.aws.amazon.com/mediastore/
2. Nella pagina Containers (Container), scegliere il nome del container.
3. Nella sezione Metric policy (Policy di parametro), scegli Edit metric policy (Modifica policy di parametro).
4. Apportare le opportune modifiche e selezionare Save (Salva).

Policy di parametro di esempio

Gli esempi seguenti mostrano policy di parametro destinate a diversi casi d'uso.

Argomenti

- Policy di parametro di esempio: parametri a livello di container
- Policy di parametro di esempio: parametri a livello di percorso
- Policy di parametro di esempio: parametri a livello di container e percorso
- Policy di parametro di esempio: parametri a livello di percorso utilizzando caratteri jolly
- Policy di parametro di esempio: parametri a livello di percorso con regole sovrapposte

Policy di parametro di esempio: parametri a livello di container

Questo esempio di policy indica che AWS Elemental MediaStore deve inviare i parametri ad Amazon a CloudWatch livello di container. Ad esempio, include il parametro `RequestCount` che conta il numero di richieste `Put` effettuate al container. In alternativa, puoi impostare su `DISABLED`.

Poiché non ci sono regole in questa policy, MediaStore non invia metriche a livello di percorso. Ad esempio, non puoi visualizzare quante richieste `Put` sono state effettuate a una determinata cartella all'interno di questo container.

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

Policy di parametro di esempio: parametri a livello di percorso

Questo esempio di policy indica che AWS Elemental non MediaStore deve inviare parametri ad Amazon a CloudWatch livello di container. Inoltre, MediaStore non deve inviare parametri per gli oggetti in due cartelle specifiche: `baseball/saturday` e `football/saturday`. I parametri per le richieste di MediaStore sono i seguenti:

- Le richieste alla `baseball/saturday` cartella hanno una CloudWatch dimensione di. `ObjectGroupName=baseballGroup`
- Le richieste alla cartella `football/saturday` hanno una dimensione `ObjectGroupName=footballGroup`.

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

Policy di parametro di esempio: parametri a livello di container e percorso

Questo esempio di policy indica che AWS Elemental MediaStore deve inviare i parametri ad Amazon a CloudWatch livello di container. Inoltre, MediaStore deve inviare i parametri per gli oggetti in due cartelle specifiche: e. `baseball/saturday` `football/saturday` I parametri per le richieste di MediaStore sono i seguenti:

- Le richieste alla `baseball/saturday` cartella hanno una CloudWatch dimensione di`ObjectGroupName=baseballGroup`.
- Le richieste alla `football/saturday` cartella hanno una CloudWatch dimensione`ObjectGroupName=footballGroup`.

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

Policy di parametro di esempio: parametri a livello di percorso utilizzando caratteri jolly

Questo esempio di policy indica che AWS Elemental MediaStore deve inviare i parametri ad Amazon a CloudWatch livello di container. Inoltre, MediaStore deve inviare anche i parametri per gli oggetti in base al nome del file. Un carattere jolly indica che gli oggetti possono essere archiviati in qualsiasi punto del container e avere qualsiasi nome del file purché terminino con un'estensione `.m3u8`.

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

Policy di parametro di esempio: parametri a livello di percorso con regole sovrapposte

Questo esempio di policy indica che AWS Elemental MediaStore deve inviare i parametri ad Amazon a CloudWatch livello di container. Inoltre, MediaStore dovrebbe inviare i parametri per due cartelle: e. `sports/football/saturday` `sports/football`

Le metriche per MediaStore le richieste alla `sports/football/saturday` cartella hanno una CloudWatch dimensione di. `ObjectGroupName=footballGroup1` Poiché gli oggetti archiviati nella cartella `sports/football` corrispondono a entrambe le regole, CloudWatch visualizza due punti

dati per questi oggetti: uno con una dimensione `ObjectGroupName=footballGroup1` e il secondo con una dimensione `ObjectGroupName=footballGroup2`.

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

Cartelle in AWS Elemental MediaStore

Le cartelle sono divisioni all'interno di un container, utilizzate per suddividere il container proprio come si fa con le sottocartelle per dividere una cartella in un file system. È possibile creare fino a 10 livelli di cartelle (escluso il container stesso).

Le cartelle sono facoltative; è possibile scegliere di caricare gli oggetti direttamente in un container, invece che in una cartella. Tuttavia, le cartelle sono un modo semplice per organizzare gli oggetti.

Per caricare un oggetto in una cartella, devi specificare il percorso della cartella. Se la cartella esiste già, AWS Elemental MediaStore memorizza l'oggetto nella cartella. Se la cartella non esiste, il servizio la crea e quindi archivia l'oggetto nella cartella.

Ad esempio, supponiamo di avere un contenitore denominato `movies` e di caricare un file denominato `mlaw.ts` con il percorso. `premium/canada` AWS Elemental MediaStore archivia l'oggetto nella sottocartella canada all'interno della cartella premium. Se nessuna delle due cartelle esiste, il servizio crea sia la cartella `premium` che la sottocartella `canada`, quindi archivia l'oggetto nella sottocartella `canada`. Se specifichi solo il container `movies` (senza percorso), il servizio archivia l'oggetto direttamente nel container.

AWS Elemental elimina MediaStore automaticamente una cartella quando elimini l'ultimo oggetto in quella cartella. e anche le eventuali cartelle superiori vuote. Ad esempio, supponi di avere una cartella denominata "premium" che non contiene file ma una sottocartella denominata `canada`. La sottocartella `canada` contiene un file denominato `mlaw.ts`. Se elimini il file `mlaw.ts`, il servizio elimina entrambe le cartelle `premium` e `canada`. L'eliminazione automatica si applica solo per le cartelle. Il servizio non elimina i container vuoti.

Argomenti

Regole per i nomi di cartella

Quando scegli un nome per la cartella, ricordati quanto segue:

- Il nome può contenere solo i seguenti caratteri: lettere maiuscole (A-Z), lettere minuscole (a-z), numeri (0-9), punti (.), trattini (-), tilde (~), trattini bassi (_), segni di uguaglianza (=) e due punti (:).
- Il nome deve contenere almeno un carattere. I nomi di cartelle vuote (ad esempio`folder1//folder3/`) non sono consentiti.
- I nomi rispettano la distinzione tra lettere maiuscole e minuscole. Ad esempio, puoi avere una cartella denominata `myFolder` e una denominata `myfolder` nello stesso container o cartella perché tali nomi sono univoci.
- Il nome deve essere univoco solo all'interno della cartella o del container padre. Ad esempio puoi creare una cartella denominata `myfolder` in due diversi container: `movies/myfolder` e `sports/myfolder`.
- Il nome può avere lo stesso nome del container padre.
- La cartella non può essere rinominata dopo che è stata creata.

Creazione di una cartella

Puoi creare le cartelle al momento di caricare gli oggetti. Per caricare un oggetto in una cartella, devi specificare il percorso della cartella. Se la cartella esiste già, AWS Elemental MediaStore memorizza l'oggetto nella cartella. Se la cartella non esiste, il servizio la crea e quindi archivia l'oggetto nella cartella.

Per ulteriori informazioni, consulta the section called “Caricamento di un oggetto”.

Eliminazione di una cartella

Puoi eliminare le cartelle solo se sono vuote; non è possibile eliminare cartelle che contengono oggetti.

AWS Elemental elimina MediaStore automaticamente una cartella quando elimini l'ultimo oggetto in quella cartella. e anche le eventuali cartelle superiori vuote. Ad esempio, supponi di avere una cartella denominata `premium` che non contiene file ma una sottocartella denominata `canada`. La sottocartella `canada` contiene un file denominato `mlaw.ts`. Se elimini il file `mlaw.ts`, il servizio elimina entrambe le cartelle `premium` e `canada`. L'eliminazione automatica si applica solo per le cartelle. Il servizio non elimina i container vuoti.

Per ulteriori informazioni, consulta Eliminazione di un oggetto.

Oggetti in AWS Elemental MediaStore

Gli MediaStore asset AWS Elemental sono chiamati oggetti. Puoi caricare un oggetto in un container o in una cartella all'interno del container.

In MediaStore, puoi caricare, scaricare ed eliminare oggetti:

- Upload (Carica): aggiungere un oggetto a un container o una cartella. Non corrisponde alla creazione di un oggetto. È necessario creare gli oggetti localmente prima di caricarli su MediaStore.
- Scarica: copia un oggetto MediaStore da un'altra posizione. Questa operazione non rimuove l'oggetto da MediaStore.
- Delete (Elimina): rimuovere completamente un oggetto da MediaStore. È possibile eliminare gli oggetti individualmente oppure aggiungere una policy del ciclo di vita degli oggetti per eliminare automaticamente gli oggetti all'interno di un container dopo un intervallo di tempo specificato.

MediaStore accetta tutti i tipi di file.

Argomenti

Caricamento di un oggetto

Puoi caricare gli oggetti in un container o in una cartella all'interno di un container. Per caricare un oggetto in una cartella, devi specificare il percorso della cartella. Se la cartella esiste già, AWS Elemental MediaStore memorizza l'oggetto nella cartella. Se la cartella non esiste, il servizio la crea e quindi archivia l'oggetto nella cartella. Per ulteriori informazioni sulle cartelle, consulta Cartelle in AWS Elemental MediaStore.

Puoi usare la MediaStore console o AWS CLI caricare oggetti.

MediaStore supporta il trasferimento di oggetti in blocchi, che riduce la latenza rendendo un oggetto disponibile per il download mentre è ancora in fase di caricamento. Per usare questa funzionalità, imposta la disponibilità di caricamento dell'oggetto su `streaming`. Puoi impostare il valore di questa intestazione quando carichi l'oggetto utilizzando l'API. Se non specificate questa intestazione nella richiesta, MediaStore assegna il valore predefinito di `standard` for the upload availability dell'oggetto.

Le dimensioni dell'oggetto non possono superare 25 MB per disponibilità di caricamento standard e a 10 MB per disponibilità di caricamento in streaming.

> **Note**
>
> I nomi dei file degli oggetti possono contenere solo lettere, numeri, punti (.), caratteri di sottolineatura (_), tilde (~), trattini (-), segni di uguale (=) e due punti (:).

Per caricare un oggetto (console)

1. Apri la console all'indirizzo. MediaStore https://console.aws.amazon.com/mediastore/
2. Nella pagina Containers (Container) scegliere il nome del container. Viene visualizzato il pannello dei dettagli del container.
3. Scegli Upload object (Carica oggetto).
4. In Target path (Percorso di destinazione) digita un percorso per le cartelle. Ad esempio, `premium/canada`. Se una delle cartelle del percorso specificato non esiste ancora, il servizio la crea automaticamente.
5. Nella sezione Object (Oggetto) scegli Browse (Sfoglia).
6. Passa alla cartella appropriata e scegli un oggetto da caricare.
7. Seleziona Open (Apri), quindi Upload (Carica).

> **Note**
>
> Se un file con lo stesso nome esiste già nella cartella selezionata, il servizio sostituisce il file originale con il file caricato.

Per caricare un oggetto (AWS CLI)

- In AWS CLI, usa il `put-object` comando. È anche possibile includere i seguenti parametri: `content-type`, `cache-control` (per consentire al chiamante di controllare il comportamento della cache dell'oggetto) e `path` (per inserire l'oggetto in una cartella all'interno del container).

> **Note**
>
> Dopo aver caricato l'oggetto, non è possibile modificare `content-type`, `cache-control` o `path`.

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

L'esempio seguente mostra il valore restituito:

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

Visualizzazione di un elenco di oggetti

Puoi utilizzare la MediaStore console AWS Elemental per visualizzare gli elementi (oggetti e cartelle) archiviati nel livello più alto di un contenitore o in una cartella. Gli elementi archiviati in una sottocartella del container o della cartella corrente non verranno visualizzati. Puoi utilizzare il AWS CLI per visualizzare un elenco di oggetti e cartelle all'interno di un contenitore, indipendentemente dal numero di cartelle o sottocartelle presenti nel contenitore.

Per visualizzare un elenco di oggetti in un determinato container (console)

1. Apri la MediaStore console all'indirizzo https://console.aws.amazon.com/mediastore/.

2. Nella pagina Containers (Container), scegli il nome del container che contiene la cartella che desideri visualizzare.
3. Scegli il nome della cartella dall'elenco.

   Viene visualizzata una pagina di dettagli che mostra tutte le cartelle e gli oggetti memorizzati nella cartella.

Per visualizzare un elenco di oggetti in una determinata cartella (console)

1. Apri la MediaStore console all'indirizzo https://console.aws.amazon.com/mediastore/.
2. Nella pagina Containers (Container), scegli il nome del container che contiene la cartella che desideri visualizzare.

   Viene visualizzata una pagina di dettagli che mostra tutte le cartelle e gli oggetti memorizzati nel container.

Per visualizzare un elenco di oggetti e cartelle in un determinato container (AWS CLI)

- In AWS CLI, usa il `list-items` comando:

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

  L'esempio seguente mostra il valore restituito:

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

```
        }
    ]
}
```

> **Note**
>
> Gli oggetti soggetti a una regola `seconds_since_create` non sono inclusi nella risposta di `list-items`.

Per visualizzare un elenco di oggetti e cartelle in una determinata cartella (AWS CLI)

- Nel AWS CLI, usa il `list-items` comando, con il nome della cartella specificato alla fine della richiesta:

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

L'esempio seguente mostra il valore restituito:

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

> **Note**
>
> Gli oggetti soggetti a una regola `seconds_since_create` non sono inclusi nella risposta di `list-items`.

Visualizzazione dei dettagli di un oggetto

Dopo aver caricato un oggetto, AWS Elemental MediaStore memorizza dettagli come la data di modifica, la lunghezza del contenuto, ETag (tag di entità) e il tipo di contenuto. Per informazioni sull'utilizzo dei metadati di un oggetto, consulta Interazione di MediaStore con le cache HTTP.

Per visualizzare i dettagli di un oggetto (console)

1. Apri la MediaStore console all'indirizzo https://console.aws.amazon.com/mediastore/.
2. Nella pagina Containers (Container), scegli il nome del container che contiene l'oggetto che desideri visualizzare.
3. Se l'oggetto che desideri visualizzare si trova in una cartella, continua a selezionare i nomi di cartella fino a visualizzare l'oggetto.
4. Scegli il nome dell'oggetto.

   Viene visualizzata una pagina di dettagli che mostra le informazioni sull'oggetto.

Per visualizzare i dettagli di un oggetto (AWS CLI)

- In AWS CLI, usa il `describe-object` comando:

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

L'esempio seguente mostra il valore restituito:

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

```
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
}
```

Download di un oggetto

È possibile utilizzare la console per scaricare un oggetto. Puoi usarla AWS CLI per scaricare un oggetto o solo una parte di un oggetto.

Per scaricare un oggetto (console)

1. Apri la MediaStore console all'indirizzo https://console.aws.amazon.com/mediastore/.
2. Nella pagina Containers (Container), scegli il nome del container che contiene l'oggetto da scaricare.
3. Se l'oggetto che desideri scaricare si trova in una cartella, continua a selezionare i nomi di cartella fino a visualizzare l'oggetto.
4. Scegli il nome dell'oggetto.
5. Nella pagina dei dettagli Object (Oggetto), scegli Download (Scarica).

Per scaricare un oggetto (AWS CLI)

- In AWS CLI, usa il `get-object` comando:

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

L'esempio seguente mostra il valore restituito:

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

Per scaricare parte di un oggetto (AWS CLI)

- In AWS CLI, usa il `get-object` comando e specifica un intervallo.

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

L'esempio seguente mostra il valore restituito:

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

Eliminazione di oggetti

AWS Elemental MediaStore offre diverse opzioni per l'eliminazione di oggetti dai contenitori:

- Elimina un singolo oggetto. Non soggetto a costi.
- Svuota un container per eliminare tutti gli oggetti all'interno del container contemporaneamente. Poiché questo processo utilizza chiamate API, si applicano le normali tariffe API.
- Aggiungi una policy del ciclo di vita degli oggetti per eliminare gli oggetti dopo un determinato periodo. Non soggetto a costi.

Eliminazione di un oggetto

Puoi eliminare gli oggetti utilizzando la console o AWS CLI. In alternativa, puoi aggiungere una policy del ciclo di vita degli oggetti per eliminare automaticamente gli oggetti dopo un determinato periodo in un container oppure svuotare un container per eliminare tutti gli oggetti all'interno del container.

> **Note**
>
> Quando elimini l'unico oggetto in una cartella, AWS Elemental elimina MediaStore automaticamente la cartella e tutte le cartelle vuote al di sopra di quella cartella. Ad esempio, supponi di avere una cartella denominata `premium` che non contiene file ma una sottocartella denominata `canada`. La sottocartella `canada` contiene un file denominato `mlaw.ts`. Se elimini il file `mlaw.ts`, il servizio elimina entrambe le cartelle `premium` e `canada`.

Per eliminare un oggetto (console)

1. Apri la MediaStore console all'indirizzo. https://console.aws.amazon.com/mediastore/
2. Nella pagina Containers (Container), scegli il nome del container che contiene l'oggetto da eliminare.
3. Se l'oggetto che desideri eliminare si trova in una cartella, continua a selezionare i nomi di cartella fino a visualizzare l'oggetto.
4. Scegli l'opzione a sinistra del nome dell'oggetto.
5. Scegliere Delete (Elimina).

Per eliminare un oggetto (AWS CLI)

- In AWS CLI, usa il `delete-object` comando.

  Esempio:

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

  Il comando non ha un valore restituito.

Svuotamento di un container

Puoi svuotare un container per eliminare tutti gli oggetti archiviati all'interno del container. In alternativa, puoi aggiungere una policy del ciclo di vita degli oggetti per eliminare automaticamente gli oggetti dopo un determinato periodo in un container oppure eliminare gli oggetti singolarmente.

Per svuotare un container (console)

1. Apri la MediaStore console all'indirizzo https://console.aws.amazon.com/mediastore/.
2. Nella pagina Containers (Container), scegli l'opzione per il container da svuotare.
3. Scegli Empty container (Svuota container). Viene visualizzato un messaggio di conferma.
4. Conferma di voler svuotare il contenitore inserendo il nome del contenitore nel campo di testo, quindi scegli Vuoto.

Sicurezza in AWS Elemental MediaStore

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te e te. AWS Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi in Cloud AWS. AWS fornisce inoltre servizi che è possibile utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per ulteriori informazioni sui programmi di conformità che si applicano ad AWS Elemental MediaStore, consulta AWS Services in Scope by Compliance Program AWS Services in Scope Program.
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. L'utente è anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della propria azienda e le leggi e normative vigenti.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa durante l'utilizzo MediaStore. Negli argomenti seguenti viene illustrato come eseguire la configurazione MediaStore per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche a utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere MediaStore le tue risorse.

Argomenti

Protezione dei dati in AWS Elemental MediaStore

Il modello di responsabilità AWS condivisa Modello si applica alla protezione dei dati in AWS Elemental MediaStore. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per maggiori informazioni sulla privacy dei dati, consulta le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al AWS Modello di responsabilità condivisa e GDPR nel AWS Blog sulla sicurezza.

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta Lavorare con i CloudTrail percorsi nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-3.

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori MediaStore o Servizi AWS utilizzi la console, l'API o. AWS CLI AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando si fornisce un URL a un server esterno, suggeriamo

vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la richiesta al server.

Crittografia dei dati

MediaStore crittografa contenitori e oggetti inattivi utilizzando l'algoritmo AES-256 standard del settore. Ti consigliamo di utilizzare MediaStore per proteggere i tuoi dati nei seguenti modi:

- Crea una politica del contenitore per controllare i diritti di accesso a tutte le cartelle e gli oggetti in quel contenitore. Per ulteriori informazioni, consulta the section called "Policy di container".
- Crea una politica di condivisione delle risorse tra origini (CORS) per consentire l'accesso selettivo tra origini diverse alle tue risorse. MediaStore Con CORS, puoi consentire alle applicazioni Web client caricate in un dominio di interagire con le risorse situate in un dominio differente. Per ulteriori informazioni, consulta the section called "Policy CORS".

Identity and Access Management per AWS Elemental MediaStore

AWS Identity and Access Management (IAM) è un software Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse. MediaStore IAM è uno Servizio AWS strumento che puoi utilizzare senza costi aggiuntivi.

Argomenti

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia in base al tuo ruolo:

- Utente del servizio: richiedi le autorizzazioni all'amministratore se non riesci ad accedere alle funzionalità (vedi Risoluzione dei problemi di MediaStore identità e accesso ad AWS Elemental)
- Amministratore del servizio: determina l'accesso degli utenti e invia le richieste di autorizzazione (vedi Come funziona AWS Elemental con MediaStore IAM)
- Amministratore IAM: scrivi policy per gestire l'accesso (vedi Esempi di policy basate sull'identità per AWS Elemental MediaStore)

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi autenticarti come utente IAM o assumendo un ruolo IAM. Utente root dell'account AWS

Puoi accedere come identità federata utilizzando credenziali provenienti da una fonte di identità come AWS IAM Identity Center (IAM Identity Center), autenticazione Single Sign-On o credenziali. Google/ Facebook Per maggiori informazioni sull'accesso, consultare la sezione Come accedere a Account AWS nella Guida per l'utente di Accedi ad AWS .

Per l'accesso programmatico, AWS fornisce un SDK e una CLI per firmare crittograficamente le richieste. Per ulteriori informazioni, consulta AWS Signature Version 4 per le richieste API nella Guida per l'utente IAM.

Account AWS utente root

Quando si crea un Account AWS, si inizia con un'identità di accesso denominata utente Account AWS root che ha accesso completo a tutte Servizi AWS le risorse. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono le credenziali come utente root, consulta Attività che richiedono le credenziali dell'utente root nella Guida per l'utente IAM.

Identità federata

Come procedura ottimale, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente della directory aziendale, del provider di identità Web o Directory Service che accede Servizi AWS utilizzando le credenziali di una fonte di identità. Le identità federate assumono ruoli che forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Per ulteriori informazioni, consulta Cos'è IAM Identity Center? nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un utente IAM è un'identità che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ti consigliamo di utilizzare credenziali temporanee invece di utenti IAM con credenziali a lungo termine. Per ulteriori informazioni, consulta Richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee nella Guida per l'utente IAM.

Un gruppo IAM specifica una raccolta di utenti IAM e semplifica la gestione delle autorizzazioni per gestire gruppi di utenti di grandi dimensioni. Per ulteriori informazioni, consulta Casi d'uso per utenti IAM nella Guida per l'utente IAM.

Ruoli IAM

Un ruolo IAM è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo passando da un ruolo utente a un ruolo IAM (console) o chiamando un'operazione AWS CLI o AWS API. Per ulteriori informazioni, consulta Metodi per assumere un ruolonella Guida per l'utente IAM.

I ruoli IAM sono utili per l'accesso federato degli utenti, le autorizzazioni utente IAM temporanee, l'accesso tra account, l'accesso tra servizi e le applicazioni in esecuzione su Amazon. EC2 Per maggiori informazioni, consultare Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Gestione dell'accesso tramite policy

Puoi controllare l'accesso AWS creando policy e collegandole a identità o risorse. AWS Una policy definisce le autorizzazioni quando è associata a un'identità o a una risorsa. AWS valuta queste politiche quando un preside effettua una richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per maggiori informazioni sui documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente IAM.

Utilizzando le policy, gli amministratori specificano chi ha accesso a cosa definendo quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Un amministratore IAM crea le policy IAM e le aggiunge ai ruoli, che gli utenti possono quindi assumere. Le policy IAM definiscono le autorizzazioni indipendentemente dal metodo utilizzato per eseguirle.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile collegare a un'identità (utente, gruppo o ruolo). Tali policy controllano le operazioni autorizzate per l'identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come creare una policy basata su identità, consultare Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Le policy basate sull'identità possono essere policy in linea (incorporate direttamente in una singola identità) o policy gestite (policy autonome collegate a più identità). Per informazioni su come scegliere tra una policy gestita o una policy inline, consultare Scelta fra policy gestite e policy inline nella Guida per l'utente IAM.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi includono le policy di trust dei ruoli IAM e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. In una policy basata sulle risorse è obbligatorio specificare un'entità principale.

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi che possono impostare le autorizzazioni massime concesse dai tipi di policy più comuni:

- Limiti delle autorizzazioni: impostano il numero massimo di autorizzazioni che una policy basata su identità ha la possibilità di concedere a un'entità IAM. Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente IAM.
- Politiche di controllo del servizio (SCPs): specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa in. AWS Organizations Per ulteriori informazioni, consultare Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations .

- Politiche di controllo delle risorse (RCPs): imposta le autorizzazioni massime disponibili per le risorse nei tuoi account. Per ulteriori informazioni, consulta Politiche di controllo delle risorse (RCPs) nella Guida per l'AWS Organizations utente.
- Le policy di sessione sono policy avanzate che si passano come parametro quando si crea una sessione temporanea per un ruolo o un utente federato. Per maggiori informazioni, consultare Policy di sessione nella Guida per l'utente IAM.

Più tipi di policy

Quando a una richiesta si applicano più tipi di policy, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle policy nella IAM User Guide.

Come funziona AWS Elemental con MediaStore IAM

Prima di utilizzare IAM per gestire l'accesso a MediaStore, scopri con MediaStore quali funzionalità IAM è disponibile l'uso.

Funzionalità IAM che puoi usare con AWS Elemental MediaStore

Funzionalità IAM	MediaStore supporto
Policy basate sull'identità	Sì
Policy basate su risorse	Sì
Operazioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione della policy (specifica del servizio)	Sì
ACLs	No
ABAC (tag nelle policy)	Parziale
Credenziali temporanee	Sì

Funzionalità IAM	MediaStore supporto
Autorizzazioni del principale	Sì
Ruoli di servizio	Sì
Ruoli collegati al servizio	No

Per avere una panoramica di alto livello su come MediaStore e altri AWS servizi funzionano con la maggior parte delle funzionalità IAM, consulta AWS i servizi che funzionano con IAM nella IAM User Guide.

Politiche basate sull'identità per MediaStore

Supporta le policy basate sull'identità: sì

Le policy basate sull'identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente di IAM.

Con le policy basate sull'identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente IAM.

Esempi di politiche basate sull'identità per MediaStore

Per visualizzare esempi di politiche basate sull' MediaStore identità, vedere. Esempi di policy basate sull'identità per AWS Elemental MediaStore

Politiche basate sulle risorse all'interno MediaStore

Supporta le policy basate sulle risorse: sì

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli di IAM e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei

servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. In una policy basata sulle risorse è obbligatorio specificare un'entità principale. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, è possibile specificare un intero account o entità IAM in un altro account come entità principale in una policy basata sulle risorse. Per ulteriori informazioni, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

> **Note**
>
> MediaStore supporta anche le politiche dei contenitori che definiscono quali entità principali (account, utenti, ruoli e utenti federati) possono eseguire azioni sul contenitore. Per ulteriori informazioni, consulta Policy di container.

Azioni politiche per MediaStore

Supporta le operazioni di policy: si

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Includere le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco di MediaStore azioni, consulta Azioni definite da AWS Elemental MediaStore nel Service Authorization Reference.

Le azioni politiche in MediaStore uso utilizzano il seguente prefisso prima dell'azione:

```
mediastore
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

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

Per visualizzare esempi di politiche MediaStore basate sull'identità, vedere. Esempi di policy basate sull'identità per AWS Elemental MediaStore

Risorse politiche per MediaStore

Supporta le risorse relative alle policy: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). Per le azioni che non supportano le autorizzazioni a livello di risorsa, utilizzare un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

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

Per visualizzare un elenco dei tipi di MediaStore risorse e relativi ARNs, consulta Resources defined by AWS Elemental MediaStore nel Service Authorization Reference. Per sapere con quali azioni è possibile specificare l'ARN di ogni risorsa, consulta Azioni definite da AWS Elemental. MediaStore

La risorsa MediaStore contenitore ha il seguente ARN:

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

Per ulteriori informazioni sul formato di ARNs, consulta Amazon Resource Names (ARNs) e AWS Service Namespaces.

Ad esempio, per specificare il container `AwardsShow` nell'istruzione, utilizza il seguente ARN:

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

Chiavi relative alle condizioni della policy per MediaStore

Supporta le chiavi di condizione delle policy specifiche del servizio: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Condition` specifica quando le istruzioni vengono eseguite in base a criteri definiti. È possibile compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Per visualizzare un elenco di chiavi di MediaStore condizione, consulta Condition keys for AWS Elemental MediaStore nel Service Authorization Reference. Per sapere con quali azioni e risorse puoi utilizzare una chiave di condizione, consulta Azioni definite da AWS Elemental MediaStore.

Per visualizzare esempi di politiche MediaStore basate sull'identità, consulta. Esempi di policy basate sull'identità per AWS Elemental MediaStore

ACLs in MediaStore

Supporti ACLs: no

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

ABAC con MediaStore

Supporta ABAC (tag nelle policy): parzialmente

Il controllo degli accessi basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi, chiamati tag. Puoi allegare tag a entità e AWS risorse IAM, quindi progettare politiche ABAC per consentire operazioni quando il tag del principale corrisponde al tag sulla risorsa.

Per controllare l'accesso basato su tag, fornire informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Sì. Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per maggiori informazioni su ABAC, consulta Definizione delle autorizzazioni con autorizzazione ABAC nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di

ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC) nella Guida per l'utente di IAM.

Utilizzo di credenziali temporanee con MediaStore

Supporta le credenziali temporanee: sì

Le credenziali temporanee forniscono l'accesso a breve termine alle AWS risorse e vengono create automaticamente quando si utilizza la federazione o si cambia ruolo. AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza temporanee in IAM e Servizi AWS compatibili con IAM nelle Guida per l'utente IAM.

Autorizzazioni principali multiservizio per MediaStore

Supporta l'inoltro delle sessioni di accesso (FAS): sì

Le sessioni di accesso inoltrato (FAS) utilizzano le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta di effettuare richieste Servizio AWS ai servizi downstream. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Inoltro sessioni di accesso.

Ruoli di servizio per MediaStore

Supporta i ruoli di servizio: sì

Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per maggiori informazioni, consulta la sezione Creare un ruolo per delegare le autorizzazioni a una persona Servizio AWS nella Guida per l'utente di IAM.

> **Warning**
>
> La modifica delle autorizzazioni per un ruolo di servizio potrebbe compromettere la funzionalità. MediaStore Modifica i ruoli di servizio solo quando viene MediaStore fornita una guida in tal senso.

Ruoli collegati ai servizi per MediaStore

Supporta i ruoli collegati ai servizi: no

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati al servizio, ma non modificarle.

Per ulteriori informazioni su come creare e gestire i ruoli collegati ai servizi, consulta Servizi AWS supportati da IAM. Trova un servizio nella tabella che include un `Yes` nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Esempi di policy basate sull'identità per AWS Elemental MediaStore

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse MediaStore. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy IAM (console) nella Guida per l'utente IAM.

Per dettagli sulle azioni e sui tipi di risorse definiti da MediaStore, incluso il formato di ARNs per ogni tipo di risorsa, consulta Azioni, risorse e chiavi di condizione per AWS Elemental MediaStore nel Service Authorization Reference.

Argomenti

- Best practice per le policy
- Utilizzo della console di MediaStore
- Consentire agli utenti di visualizzare le loro autorizzazioni

Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare MediaStore risorse nel tuo account. Queste operazioni possono comportare costi aggiuntivi per l' Account AWS. Quando si creano o modificano policy basate sull'identità, seguire queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono

le autorizzazioni per molti casi d'uso comuni.AWS Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per maggiori informazioni, consulta Policy gestite da AWS o Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente di IAM.

- Applicazione delle autorizzazioni con privilegio minimo - Quando si impostano le autorizzazioni con le policy IAM, concedere solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegio minimo. Per maggiori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso - Per limitare l'accesso a operazioni e risorse è possibile aggiungere una condizione alle policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio CloudFormation. Per maggiori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente di IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali - IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per maggiori informazioni, consulta Convalida delle policy per il Sistema di analisi degli accessi IAM nella Guida per l'utente di IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungere le condizioni MFA alle policy. Per maggiori informazioni, consulta Protezione dell'accesso API con MFA nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consultare Best practice di sicurezza in IAM nella Guida per l'utente IAM.

Utilizzo della console di MediaStore

Per accedere alla MediaStore console AWS Elemental, devi disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle MediaStore risorse del tuo. Account AWS Se si cra una policy basata sull'identità più restrittiva

rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime per la console agli utenti che effettuano chiamate solo verso AWS CLI o l' AWS API. Al contrario, è opportuno concedere l'accesso solo alle operazioni che corrispondono all'operazione API che stanno cercando di eseguire.

Per garantire che utenti e ruoli possano ancora utilizzare la MediaStore console, allega anche la policy MediaStore *ConsoleAccess* o la policy *ReadOnly* AWS gestita alle entità. Per maggiori informazioni, consulta Aggiunta di autorizzazioni a un utente nella Guida per l'utente di IAM.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono cpllegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o a livello di codice. AWS CLI AWS

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

```
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Risoluzione dei problemi di MediaStore identità e accesso ad AWS Elemental

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con MediaStore un IAM.

Argomenti

- Non sono autorizzato a eseguire alcuna azione in MediaStore
- Non sono autorizzato a eseguire iam: PassRole
- Voglio consentire a persone esterne a me di accedere Account AWS alle mie MediaStore risorse

Non sono autorizzato a eseguire alcuna azione in MediaStore

Se ricevi un errore che indica che non sei autorizzato a eseguire un'operazione, le tue policy devono essere aggiornate per poter eseguire l'operazione.

L'errore di esempio seguente si verifica quando l'utente IAM `mateojackson` prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *`my-example-widget`* fittizia ma non dispone di autorizzazioni `mediastore:`*`GetWidget`* fittizie.

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

In questo caso, la policy per l'utente `mateojackson` deve essere aggiornata per consentire l'accesso alla risorsa *`my-example-widget`* utilizzando l'azione `mediastore:`*`GetWidget`*.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un errore che indica che non sei autorizzato a eseguire l'operazione `iam:PassRole`, le tue policy devono essere aggiornate per poter passare un ruolo a MediaStore.

Alcuni Servizi AWS consentono di passare un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un utente IAM denominato `marymajor` cerca di utilizzare la console per eseguire un'operazione in MediaStore. Tuttavia, l'operazione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per trasmettere il ruolo al servizio.

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

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire a persone esterne a me di accedere Account AWS alle mie MediaStore risorse

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per concedere alle persone l'accesso alle tue risorse.

Per maggiori informazioni, consulta gli argomenti seguenti:

- Per sapere se MediaStore supporta queste funzionalità, consulta. Come funziona AWS Elemental con MediaStore IAM
- Per scoprire come fornire l'accesso alle tue risorse attraverso Account AWS le risorse di tua proprietà, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a soggetti Account AWS di proprietà di terze parti nella Guida per l'utente IAM.

- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso a utenti autenticati esternamente (Federazione delle identità) nella Guida per l'utente di IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente di IAM.

Registrazione e monitoraggio AWS Elemental MediaStore

Questa sezione fornisce una panoramica delle opzioni per il logging e il monitoraggio in AWS Elemental MediaStore per scopi di sicurezza. Per ulteriori informazioni sul logging e il monitoraggio in MediaStore consulta Monitoraggio e etichettatura in AWS Elemental MediaStore.

Il monitoraggio è un elemento importante per mantenere l'affidabilità, la disponibilità e le prestazioni delle soluzioni AWS Elemental MediaStore esistenti. AWS È necessario raccogliere i dati di monitoraggio da tutte le parti della AWS soluzione in modo da poter eseguire più facilmente il debug di un errore multipunto, se si verifica. AWS fornisce diversi strumenti per monitorare le MediaStore risorse e rispondere a potenziali incidenti.

CloudWatch Allarmi Amazon

Utilizzando gli CloudWatch allarmi, osservi una singola metrica per un periodo di tempo specificato. Se la metrica supera una determinata soglia, viene inviata una notifica a un argomento di Amazon SNS o a una policy di AWS Auto Scaling. CloudWatch gli allarmi non richiamano azioni perché si trovano in uno stato particolare. È necessario invece cambiare lo stato e mantenerlo per un numero di periodi specificato. Per ulteriori informazioni, consulta Monitoraggio con CloudWatch.

AWS CloudTrail registri

CloudTrail fornisce un registro delle azioni intraprese da un utente, un ruolo o un AWS servizio in AWS Elemental MediaStore. Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare a quale richiesta è stata inviata MediaStore, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi. Per ulteriori informazioni, consulta Registrazione delle chiamate API con CloudTrail.

AWS Trusted Advisor

Trusted Advisor si basa sulle migliori pratiche apprese servendo centinaia di migliaia di AWS clienti. Trusted Advisor ispeziona il tuo ambiente AWS e poi formula raccomandazioni quando esistono

opportunità per risparmiare denaro, migliorare la disponibilità e le prestazioni del sistema o contribuire a colmare le lacune di sicurezza. Tutti i AWS clienti hanno accesso a cinque controlli Trusted Advisor. I clienti con un piano di supporto Business o Enterprise possono visualizzare tutti i Trusted Advisor controlli.

Per ulteriori informazioni, consulta AWS Trusted Advisor.

Convalida della conformità per AWS Elemental MediaStore

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione Scope by Compliance Program Servizi AWS e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di AWS conformità Programmi di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. Per ulteriori informazioni sulla responsabilità di conformità durante l'utilizzo Servizi AWS, consulta la Documentazione AWS sulla sicurezza.

Resilienza in AWS Elemental MediaStore

L'infrastruttura AWS globale è costruita attorno Regioni AWS a zone di disponibilità. Regioni AWS forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità è possibile progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni sulle zone di disponibilità, vedere Global Regioni AWS Infrastructure.AWS

Oltre all'infrastruttura AWS globale, MediaStore offre diverse funzionalità per supportare le esigenze di resilienza e backup dei dati.

Sicurezza dell'infrastruttura in AWS Elemental MediaStore

In quanto servizio gestito, AWS Elemental MediaStore è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi di AWS sicurezza e su come AWS protegge l'infrastruttura, consulta AWS Cloud Security. Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Utilizzate chiamate API AWS pubblicate per accedere MediaStore attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Prevenzione del confused deputy tra servizi

Con "confused deputy" si intende un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire una certa operazione può costringere un'entità con più privilegi a eseguire tale operazione. Nel AWS, l'impersonificazione tra servizi può portare al confuso problema del vicesceriffo. La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua una chiamata a un altro servizio (il servizio chiamato). Il servizio chiamante può essere manipolato per utilizzare le proprie autorizzazioni e agire sulle risorse di un altro cliente, a cui normalmente non avrebbe accesso. Per evitare ciò, AWS fornisce strumenti per poterti a proteggere i tuoi dati per tutti i servizi con entità di servizio a cui è stato concesso l'accesso alle risorse del tuo account.

Consigliamo di utilizzare le chiavi di contesto `aws:SourceArn` e `aws:SourceAccount`global condition nelle policy delle risorse per limitare le autorizzazioni che AWS MediaStore Elemental fornisce a un altro servizio alla risorsa. Utilizzare `aws:SourceArn` se si desidera consentire l'associazione di una sola risorsa all'accesso tra servizi. Utilizzare `aws:SourceAccount` se si desidera consentire l'associazione di qualsiasi risorsa in tale account all'uso tra servizi.

Il modo più efficace per proteggersi dal problema "confused deputy" è quello di utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con l'ARN completo della risorsa. Se non si conosce l'ARN completo della risorsa o si scelgono più risorse, utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con caratteri jolly (*) per le parti sconosciute dell'ARN. Ad esempio, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Se il valore `aws:SourceArn` non contiene l'ID account, ad esempio un ARN di un bucket Amazon S3, è necessario utilizzare entrambe le chiavi di contesto delle condizioni globali per limitare le autorizzazioni.

Il valore di `aws:SourceArn` deve essere la configurazione per la quale vengono MediaStore pubblicati CloudWatch i log nella tua regione e nel tuo account.

L'esempio seguente mostra come utilizzare le chiavi di contesto `aws:SourceArn` e `aws:SourceAccount` global condition MediaStore per evitare il confuso problema del vice.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "ConfusedDeputyPreventionExamplePolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "mediastore.amazonaws.com"
      },
      "Action": "mediastore:CreateContainer",
      "Resource": [
        "arn:aws:mediastore:us-east-2:333333333333:container/ResourceName/*"
      ],
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:mediastore:*:333333333333:*"
        },
        "StringEquals": {
          "aws:SourceAccount": "333333333333"
        }
      }
    }
  ]
}
```

Monitoraggio e etichettatura in AWS Elemental MediaStore

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni di AWS Elemental MediaStore e delle altre AWS soluzioni. AWS fornisce i seguenti strumenti di monitoraggio per osservare MediaStore, segnalare quando qualcosa non va e intraprendere azioni automatiche quando necessario:

- AWS CloudTrailacquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo AWS account e invia i file di log a un bucket Amazon S3 da te specificato. Puoi identificare quali utenti e account hanno chiamato AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute le chiamate. Per ulteriori informazioni, consulta la Guida per l'utente AWS CloudTrail.
- Amazon CloudWatch monitora AWS le tue risorse e le applicazioni su cui esegui AWS in tempo reale. È possibile raccogliere e tenere traccia dei parametri, creare pannelli di controllo personalizzati e impostare allarmi per inviare una notifica o intraprendere azioni quando un parametro specificato raggiunge una determinata soglia. Ad esempio, puoi tenere CloudWatch traccia dell'utilizzo della CPU o di altri parametri delle tue EC2 istanze Amazon e avviare automaticamente nuove istanze quando necessario. Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.
- Amazon CloudWatch Events offre un flusso di eventi di sistema che descrivono i cambiamenti nelle AWS risorse. In genere, AWS i servizi inviano notifiche di CloudWatch eventi a Events in pochi secondi, ma a volte possono richiedere un minuto o più. CloudWatch Events consente l'elaborazione automatizzata basata sugli eventi, in quanto è possibile scrivere regole che controllano determinati eventi e attivano azioni automatiche in altri AWS servizi quando si verificano tali eventi. Per ulteriori informazioni, consulta la Amazon CloudWatch Events User Guide.
- Amazon CloudWatch Logs ti consente di monitorare, archiviare e accedere ai tuoi file di registro da EC2 istanze Amazon e altre fonti. CloudTrail CloudWatch I log possono monitorare le informazioni nei file di registro e avvisarti quando vengono raggiunte determinate soglie. Puoi inoltre archiviare i dati del log in storage estremamente durevole. Per ulteriori informazioni, consulta la Amazon CloudWatch Logs User Guide.

Puoi anche assegnare metadati ai tuoi MediaStore contenitori sotto forma di tag. Ogni tag è un'etichetta che comprende una chiave e il valore definiti. I tag possono semplificare la gestione, la ricerca e il filtro delle risorse. Puoi utilizzare i tag per organizzare AWS le risorse nella Console di

AWS gestione, creare report di utilizzo e fatturazione per tutte le tue AWS risorse e filtrare le risorse durante le attività di automazione dell'infrastruttura.

Argomenti

Registrazione delle chiamate API AWS MediaStore Elemental con AWS CloudTrail

AWS Elemental MediaStore è integrato con AWS CloudTrail un servizio che fornisce un registro delle azioni intraprese da un utente, ruolo o AWS servizio in. MediaStore CloudTrail acquisisce un sottoinsieme di chiamate API per eventi MediaStore as, incluse le chiamate dalla MediaStore console e le chiamate di codice all'API. MediaStore Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per. MediaStore Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare a quale richiesta è stata inviata MediaStore, l'indirizzo IP da cui è stata effettuata, chi ha effettuato la richiesta, quando è stata effettuata e altro ancora.

Per ulteriori informazioni CloudTrail, incluso come configurarlo e abilitarlo, consulta la Guida per l'AWS CloudTrail utente.

Argomenti

Informazioni su AWS Elemental MediaStore in CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività di evento supportata in AWS Elemental MediaStore, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi di MediaStore, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando si crea un trail nella console, il trail sarà valido in tutte le Regioni AWS . Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta i seguenti argomenti:

- Panoramica della creazione di un trail
- CloudTrail Servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

AWS Elemental MediaStore supporta la registrazione delle seguenti operazioni come eventi nei CloudTrail file di registro:

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

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente root o utente
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.

- Se la richiesta è stata effettuata da un altro AWS servizio

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Esempio: voci dei file di MediaStore log di AWS Elemental

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta da qualsiasi sorgente e include informazioni sull'operazione richiesta, la data e l'ora dell'operazione, i parametri della richiesta e così via. I file di log di CloudTrail non sono uno stack trace ordinato delle chiamate API pubbliche, pertanto queste non vengono visualizzate in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che dimostra l'`CreateContainer`operazione:

```
{
    "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"
}
```


Monitoraggio di AWS Elemental MediaStore con Amazon CloudWatch

Puoi monitorare AWS Elemental MediaStore utilizzando CloudWatch, che raccoglie dati grezzi e li elabora in metriche leggibili. CloudWatch conserva le statistiche per 15 mesi in modo da poter accedere alle informazioni storiche e avere una prospettiva migliore sulle prestazioni della tua applicazione o del tuo servizio web. È anche possibile impostare allarmi che controllano determinate soglie e inviare notifiche o intraprendere azioni quando queste soglie vengono raggiunte. Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.

AWS fornisce i seguenti strumenti di monitoraggio per osservare MediaStore, segnalare quando qualcosa non va e intraprendere azioni automatiche quando necessario:

- Amazon CloudWatch Logs ti consente di monitorare, archiviare e accedere ai tuoi file di log da AWS servizi come MediaStore AWS Elemental. Puoi usare CloudWatch Logs per monitorare applicazioni e sistemi utilizzando i dati di registro. Ad esempio, CloudWatch Logs può tenere traccia del numero di errori che si verificano nei registri delle applicazioni e inviare una notifica ogni volta che il tasso di errori supera una soglia specificata. CloudWatch Logs utilizza i dati di registro per il monitoraggio, quindi non sono necessarie modifiche al codice. Ad esempio, è possibile monitorare i registri delle applicazioni per termini letterali specifici (come "ValidationException«) o contare il numero di `PutObject` richieste effettuate durante un determinato periodo di tempo. Quando viene trovato il termine che state cercando, CloudWatch Logs riporta i dati in base a una

CloudWatch metrica specificata dall'utente. I dati di log vengono crittografati durante il transito e mentre sono a riposo.

- Amazon CloudWatch Events offre eventi di sistema che descrivono i cambiamenti nelle AWS risorse, come MediaStore gli oggetti. In genere, AWS i servizi inviano notifiche di CloudWatch eventi a Events in pochi secondi, ma a volte possono richiedere un minuto o più. Puoi impostare regole per abbinare gli eventi (come una `DeleteObject` richiesta) e indirizzarli a una o più funzioni o flussi di destinazione. CloudWatch Gli eventi vengono a conoscenza dei cambiamenti operativi man mano che si verificano. Inoltre, CloudWatch Events risponde a queste modifiche operative e intraprende le azioni correttive necessarie, inviando messaggi per rispondere all'ambiente, attivando funzioni, apportando modifiche e acquisendo informazioni sullo stato.

CloudWatch Registri

La registrazione degli accessi fornisce record dettagliati per le richieste che vengono effettuate a oggetti in un container. I log di accesso sono utili per molte applicazioni, ad esempio controlli di accesso e di sicurezza. Possono anche aiutarti a conoscere la tua base clienti e a comprendere la tua MediaStore fattura. CloudWatch I log sono classificati come segue:

- Un flusso di log è una sequenza di log eventi che condividono la stessa origine.
- Un gruppo di log è un gruppo di flussi di log che condividono le stesse impostazioni di conservazione, monitoraggio e controllo degli accessi. Quando abiliti la registrazione degli accessi su un contenitore, MediaStore crea un gruppo di log con un nome come. `/aws/mediastore/MyContainerName` Puoi definire i gruppi di log e specificare quali flussi inserire in ciascun gruppo. Non vi è alcuna quota per il numero di flussi di log che possono appartenere a un gruppo di log.

Per impostazione predefinita, i log vengono conservati a tempo indeterminato e non scadono mai. Puoi modificare la policy di conservazione per ogni gruppo di log mantenendo la conservazione a tempo indeterminato o scegliendo un periodo di conservazione da un giorno a 10 anni.

Configurazione delle autorizzazioni per Amazon CloudWatch

Usa AWS Identity and Access Management (IAM) per creare un ruolo che consenta ad AWS MediaStore Elemental di accedere ad Amazon. CloudWatch Devi eseguire questi passaggi per pubblicare CloudWatch i log per il tuo account. CloudWatch pubblica automaticamente le metriche per il tuo account.

Per consentire l'accesso MediaStore a CloudWatch

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione della console IAM, scegli Policies (Policy), quindi scegli Create policy (Crea policy).
3. Scegliere la scheda JSON e incollare la policy seguente:

   JSON

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

   Questa politica consente di MediaStore creare gruppi di log e flussi di log per qualsiasi contenitore in qualsiasi regione all'interno del tuo AWS account.
4. Scegliere Esamina policy.
5. Nella pagina Review policy (Esamina policy), in Name (Nome) immettere **`MediaStoreAccessLogsPolicy`** e quindi scegliere Create policy (Crea policy).
6. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.

7. Selezionare il tipo di ruolo Another AWS account (Un altro account AWS).
8. Per Account ID, inserisci l'ID del tuo AWS account.
9. Scegli Successivo: autorizzazioni.
10. Nella casella di ricerca immetti **`MediaStoreAccessLogsPolicy`**.
11. Selezionare la casella di controllo accanto alla nuova policy, quindi scegliere Next: Tags (Successivo: Tag).
12. Scegliere Next: Review (Successivo: Esamina) per visualizzare in anteprima i nuovi utenti.
13. In Role name (Nome ruolo) immettere **`MediaStoreAccessLogs`** e quindi selezionare Create role (Crea ruolo).
14. Nel messaggio di conferma, scegliere il nome del ruolo creato (**`MediaStoreAccessLogs`**).
15. Nella pagina Summary (Riepilogo) del ruolo, selezionare la scheda Trust relationships (Relazioni di trust).
16. Seleziona Modifica relazione di attendibilità.
17. Nel documento di policy, impostare l'entità principale sul servizio MediaStore. L'URL dovrebbe essere simile a questo:

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

La policy intera dovrebbe risultare come segue:

JSON

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

```
}
```

18. Scegli Update Trust Policy (Aggiorna policy di trust).

Abilitazione della registrazione degli accessi per un container

Per impostazione predefinita, AWS Elemental MediaStore non raccoglie i log di accesso. Quando abiliti la registrazione degli accessi su un container, MediaStore consegna i log di accesso per gli oggetti archiviati in quel contenitore ad Amazon. CloudWatch I log di accesso forniscono record dettagliati per le richieste che vengono effettuate a qualsiasi oggetto archiviato nel container. I report possono includere informazioni quali il tipo di richiesta, le risorse in essa specificate, l'ora e la data di elaborazione.

> **Important**
>
> L'attivazione di questa funzione non comporta costi aggiuntivi su un container MediaStore. Tuttavia, i file di log distribuiti dal servizio accumulano i consueti addebiti per lo storage. (Puoi eliminare il file di log in qualsiasi momento.) AWS non valuta i costi di trasferimento dati per la distribuzione di file di log, ma addebita le normali tariffe di trasferimento dati per l'accesso ai file di log.

Per abilitare la registrazione degli accessi (AWS CLI)

- Nel AWS CLI, usa il `start-access-logging` comando:

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

  Il comando non ha un valore restituito.

Disabilitazione della registrazione degli accessi per un container

Quando disabiliti la registrazione degli accessi su un contenitore, AWS MediaStore Elemental interrompe l'invio dei log di accesso ad Amazon. CloudWatch Questi log di accesso non vengono salvati e non sono recuperabili.

Per disabilitare la registrazione degli accessi (AWS CLI)

- Nel AWS CLI, usa il comando: `stop-access-logging`

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

Il comando non ha un valore restituito.

Risoluzione dei problemi di registrazione degli accessi in AWS Elemental MediaStore

Se i log di MediaStore accesso di AWS Elemental non vengono visualizzati in Amazon CloudWatch, consulta la tabella seguente per le possibili cause e soluzioni.

> **Note**
>
> Assicurati di abilitare AWS CloudTrail Logs per facilitare il processo di risoluzione dei problemi.

Caratteristiche	Il problema potrebbe essere...	Prova questa soluzione...
Non viene visualizzato alcun CloudTrai l evento, anche se CloudTrail i log sono abilitati.	Il ruolo IAM non esiste o il nome, le autorizzazioni o la policy di attendibilità non sono corretti.	Crea un ruolo con il nome, le autorizzazioni e la policy di attendibilità corrette. Per informazioni, consulta the section called "Configur azione delle autorizzazioni per CloudWatch".
Hai inviato una richiesta API `DescribeContainer` , ma la risposta mostra che il parametro `AccessLog gingEnabled` ha un valore di `False`. Inoltre, non visualizzi eventi CloudTrail per il ruolo `MediaStor eAccessLogs` quando effettui una chiamata `DescribeLogGroup` , `CreateLogGroup` , `DescribeL ogStream` o `CreateLogStream` .	Il ruolo IAM non esiste o il nome, le autorizzazioni o la policy di attendibilità non sono corretti.	Crea un ruolo con il nome, le autorizzazioni e la policy di attendibilità corrette. Per informazioni, consulta the section called "Configur azione delle autorizzazioni per CloudWatch".

Caratteristiche	Il problema potrebbe essere...	Prova questa soluzione...
	La registrazione degli accessi non è abilitata sul container.	Abilita i log di accesso per il container. Per informazioni, consulta the section called "Abilitazione della registrazione degli accessi".

Sulla CloudTrail console, viene visualizzato un evento con un errore di accesso negato relativo al `MediaStoreAccessLogs` ruolo. L' CloudTrail evento potrebbe includere righe come le seguenti:
`"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:",`
	Il ruolo IAM non dispone delle autorizzazioni corrette per MediaStore AWS Elemental.	Aggiorna il ruolo IAM per avere le autorizzazioni e la policy di attendibilità corretti. Per informazioni, consulta the section called "Configurazione delle autorizzazioni per CloudWatch".

Caratteristiche	Il problema potrebbe essere...	Prova questa soluzione...
Non vedi alcun log per un intero container o più container.	Il tuo account potrebbe aver superato la CloudWatch quota di gruppi di log per account per regione. Consulta le quote per i gruppi di log nella Amazon CloudWatch Logs User Guide.	Sulla CloudWatch console, determina se il tuo account ha raggiunto la CloudWatch quota per i gruppi di log. Se necessario, richiedere un aumento delle quote.
Vengono visualizzati alcuni accessi CloudWatch, ma non tutti i registri che si prevede di visualizzare.	Il tuo account potrebbe aver superato la CloudWatch quota di transazioni al secondo per account per regione. Consulta le quote `PutLogEvents` nella Amazon CloudWatch Logs User Guide.	Richiedi un aumento della quota di CloudWatch transazioni al secondo per account e regione.

Formato del log di accesso

I file di log di accesso sono costituiti da una sequenza di record di log in formato JSON, dove ogni record di log rappresenta una richiesta. L'ordine dei campi all'interno del log può variare. Di seguito è riportato un esempio di log costituito da due record di 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
}
```

L'elenco di seguito descrive i campi dei record di log.

AWSAccountId

L' AWS ID dell'account utilizzato per effettuare la richiesta.

BytesReceived

Il numero di byte nel corpo della richiesta che il server MediaStore riceve.

BytesSent

Il numero di byte nel corpo della risposta inviato dal server MediaStore. Tale valore spesso è identico a quello dell'intestazione `Content-Length` inclusa con le risposte del server.

ContainerName

Il nome del container che ha ricevuto la richiesta.

ErrorCode

Il codice MediaStore di errore (ad esempio`InternalServerError`). Se non si è verificato alcun errore, viene visualizzato il carattere `-`. Un codice di errore può essere visualizzato anche se il codice di stato è 200 (che indica una connessione chiusa o un errore dopo che il server ha avviato lo streaming della risposta).

ExpiresAt

Data e ora di scadenza dell'oggetto. Questo valore si basa sull'età di scadenza impostata da una transient data rule politica del ciclo di vita applicata al contenitore. Il valore è la data e ora `ISO-8601` ed è basata sull'orologio di sistema dell'host che ha servito la richiesta. Se la politica del ciclo di vita non dispone di una regola sui dati transitori che si applica all'oggetto o se non è applicata alcuna politica sul ciclo di vita al contenitore, il valore di questo campo è. `null` Questo campo si applica solo alle seguenti operazioni:, e. `PutObject` `GetObject` `DescribeObject` `DeleteObject`

HTTPStatus

Il codice di stato HTTP numerico della risposta.

Operazioni

L'operazione che è stata eseguita, ad esempio `PutObject` o `ListItems`.

Path

Il percorso all'interno del container in cui è archiviato l'oggetto. Se l'operazione non accetta un parametro path, viene visualizzato il carattere `-`.

ReceivedTime

L'ora del giorno in cui la richiesta è stata ricevuta. Il valore è la data e ora `ISO-8601` ed è basata sull'orologio di sistema dell'host che ha servito la richiesta.

Richiedente

L'Amazon Resource Name (ARN) dell'utente dell'account che è stato utilizzato per effettuare la richiesta. Per le richieste non autenticate, questo valore è `anonymous`. Se la richiesta non riesce prima del completamento dell'autenticazione, questo campo potrebbe mancare dal registro. Per tali richieste, `ErrorCode` potrebbe identificare il problema di autorizzazione.

RequestID

Una stringa generata da AWS MediaStore Elemental per identificare in modo univoco ogni richiesta.

Origine

L'indirizzo Internet apparente del richiedente o l'entità principale del servizio AWS che effettua la chiamata. Se proxy e firewall intermedi oscurano l'indirizzo del computer che effettua la richiesta, il valore è impostato su null.

TotalTime

Il numero di millisecondi (ms) durante i quali la richiesta è stata in transito dalla prospettiva del server. Tale valore viene misurato dal momento in cui la richiesta viene ricevuta dal servizio, fino al momento in cui l'ultimo byte della risposta è stato inviato. Questo valore viene misurato dalla prospettiva del server perché misurazioni effettuate dalla prospettiva del client non sono influenzate dalla latenza di rete.

TurnAroundTime

Il numero di millisecondi impiegato per l'elaborazione della richiesta. MediaStore Questo valore viene misurato dal momento in cui si riceve l'ultimo byte della richiesta al momento in cui viene inviato il primo byte di risposta.

L'ordine dei campi nel log può variare.

Tempo richiesto per l'applicazione delle modifiche dello stato di registrazione

L'applicazione effettiva delle modifiche dello stato di registrazione di un container sulla distribuzione dei file di log richiede tempo. Ad esempio, se si abilita la registrazione per un container, è possibile che nell'ora successiva alcune richieste vengano registrate nel log e altre no. Se si disabilita la registrazione per container B, alcuni log per l'ora successiva potrebbero continuare a essere recapitati, mentre altri no. In tutti i casi, le nuove impostazioni diventano effettive automaticamente.

Distribuzione dei log del server sulla base del miglior tentativo

I report dei log di accesso vengono distribuiti sulla base del miglior tentativo. La maggior parte delle richieste di un container correttamente configurato per la registrazione determinano la distribuzione di un record del log. La maggior parte dei record di log viene consegnata entro poche ore dal momento della creazione, ma possono essere consegnati con maggior frequenza.

La completezza e la tempestività della registrazione degli accessi non è tuttavia garantita. È possibile che il report del log per una richiesta specifica venga consegnato molto tempo dopo l'elaborazione effettiva della richiesta o non venga consegnato affatto. Lo scopo dei log di accesso è fornire un'idea della natura del traffico nel container. I report del log vengono persi raramente, ma la registrazione degli accessi non intende essere un resoconto completo di tutte le richieste.

Il fatto che la funzione di registrazione degli accessi si basi sul miglior tentativo fa sì che i report di utilizzo disponibili nel portale AWS (report Gestione di costi e fatturazione nella Console di gestione AWS) possano includere una o più richieste di accesso non visibili nel log di accesso distribuito.

Considerazioni in materia di programmazione per il formato dei log di accesso

Di tanto in tanto, è possibile estendere il formato dei log di accesso aggiungendo nuovi campi. Il codice che analizza i log di accesso deve essere scritto per gestire ulteriori campi che non capisce.

CloudWatch Eventi

Amazon CloudWatch Events ti consente di automatizzare AWS i tuoi servizi e rispondere automaticamente a eventi di sistema come problemi di disponibilità delle applicazioni o modifiche delle risorse. Puoi compilare regole semplici che indichino quali eventi sono considerati di interesse per te e quali azioni automatizzate intraprendere quando un evento corrisponde a una regola.

> **Important**
>
> In genere, AWS i servizi inviano notifiche di CloudWatch eventi a Events in pochi secondi, ma a volte possono richiedere un minuto o più.

Quando un file viene caricato in un contenitore o rimosso da un contenitore, nel CloudWatch servizio vengono generati due eventi in successione:

1. the section called "Evento di modifica dello stato di un oggetto"
2. the section called "Evento di modifica dello stato di un container"

Per informazioni sulla sottoscrizione a questi eventi, consulta Amazon CloudWatch.

Le azioni che possono essere attivate automaticamente includono le seguenti:

- Invocare una funzione AWS Lambda
- Richiamo del comando Amazon EC2 Run
- Inoltro dell'evento a Amazon Kinesis Data Streams
- Attivazione di una macchina a stati AWS Step Functions
- Notifica di un argomento o di una coda di Amazon SNS AWS SMS

Alcuni esempi di utilizzo di CloudWatch Events con AWS Elemental MediaStore includono quanto segue:

- Attivazione di una funzione Lambda ogni volta che viene creato un contenitore
- Notifica di un argomento di Amazon SNS quando un oggetto viene eliminato

Per ulteriori informazioni, consulta la Amazon CloudWatch Events User Guide.

Argomenti

- Evento di modifica dello stato degli MediaStore oggetti AWS Elemental
- Evento di modifica dello stato del MediaStore contenitore AWS Elemental

Evento di modifica dello stato degli MediaStore oggetti AWS Elemental

Questo evento viene pubblicato quando lo stato di un oggetto cambia (quando l'oggetto è stato caricato o eliminato).

> **Note**
>
> Gli oggetti che scadono a causa di una regola transitoria sui dati non emettono un CloudWatch evento quando scadono.

Per informazioni sull'iscrizione a questo evento, consulta Amazon CloudWatch.

Oggetto aggiornato

```
{
  "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"
  }
}
```

Oggetto rimosso

```
{
  "version": "1",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "MediaStore Object State Change",
  "source": "aws.mediastore",
  "account": "111122223333",
  "time": "2017-02-22T18:43:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:mediastore:us-east-1:111122223333:Movies/MondayMornings/Episode1/
Introduction.avi"
  ],
  "detail": {
    "ContainerName": "Movies",
    "Operation": "REMOVE",
    "Path":"Movies/MondayMornings/Episode1/Introduction.avi",
    "URL": "https://a832p1qeaznlp9.files.mediastore-us-west-2.com/Movies/
MondayMornings/Episode1/Introduction.avi"
  }
```

```
}
```

Evento di modifica dello stato del MediaStore contenitore AWS Elemental

Questo evento viene pubblicato quando lo stato di un container cambia (quando il container è stato aggiunto o eliminato). Per informazioni sull'iscrizione a questo evento, consulta Amazon CloudWatch.

Container creato

```
{
  "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"
  }
}
```

Container rimosso

```
{
  "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"
  }
}
```

Monitoraggio di AWS Elemental con i parametri di MediaStore Amazon CloudWatch

Puoi monitorare AWS Elemental MediaStore utilizzando CloudWatch, che raccoglie dati grezzi e li elabora in metriche leggibili. CloudWatchmantiene le statistiche per 15 mesi in modo da poter accedere alle informazioni storiche e avere una prospettiva migliore sulle prestazioni della tua applicazione o del tuo servizio web. È anche possibile impostare allarmi che controllano determinate soglie e inviare notifiche o intraprendere azioni quando queste soglie vengono raggiunte. Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.

Per AWS Elemental MediaStore, potresti voler controllare `BytesDownloaded` e inviare un'e-mail a te stesso quando tale metrica raggiunge una certa soglia.

Per visualizzare le metriche utilizzando la console CloudWatch

I parametri vengono raggruppati prima in base allo spazio dei nomi del servizio e successivamente in base alle diverse combinazioni di dimensioni all'interno di ogni spazio dei nomi.

1. Accedi a Console di gestione AWS e apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.
3. In Tutte le metriche, scegli lo spazio dei nomi AWS/ MediaStore.
4. Scegli la dimensione del parametro per visualizzare i parametri. Ad esempio, seleziona `Request metrics by container` per visualizzare i parametri per i diversi tipi di richieste inviate al container.

Per visualizzare le metriche utilizzando il AWS CLI

- Al prompt dei comandi, utilizza il comando seguente:

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

Metriche di AWS Elemental MediaStore

La tabella seguente elenca i parametri a cui invia AWS MediaStore Elemental. CloudWatch

> **Note**
>
> Per visualizzare le metriche, devi aggiungere una politica delle metriche al contenitore per consentire l'invio di metriche MediaStore ad Amazon. CloudWatch

Metrica	Description
`RequestCount`	Il numero totale di richieste HTTP effettuate a un container MediaStore, separate dal tipo di operazione (`Put`, `Get`, `Delete`, `Describe`, `List`). Unità: numero Dimensioni valide: • Nome del container • Nome del gruppo di oggetti • Tipo richiesta Statistiche valide: somma
`4xxErrorCount`	Il numero di richieste HTTP effettuate in tal senso ha generato un errore 4xx. MediaStore Unità: numero Dimensioni valide: • Nome del container • Nome del gruppo di oggetti • Tipo richiesta Statistiche valide: somma

Metrica	Description
`5xxErrorCount`	Il numero di richieste HTTP effettuate a MediaStore tale scopo ha generato un errore 5xx. Unità: numero Dimensioni valide: • Nome del container • Nome del gruppo di oggetti • Tipo richiesta Statistiche valide: somma
`BytesUploaded`	Il numero di byte caricati per le richieste effettuate a un container MediaStore in cui la richiesta include un corpo. Unità: byte Dimensioni valide: • Nome del container • Nome del gruppo di oggetti Statistiche valide: media (byte per richiesta), somma (byte per periodo), numero di esempi, minimo (come P0,0), massimo (come p100), qualsiasi percentile tra p0,0 e p99,9

Metrica	Description
`BytesDownloaded`	Il numero di byte scaricati per le richieste effettuate a un container MediaStore in cui la risposta include un corpo. Unità: byte Dimensioni valide: • Nome del container • Nome del gruppo di oggetti Statistiche valide: media (byte per richiesta), somma (byte per periodo), numero di esempi, minimo (come P0,0), massimo (come p100), qualsiasi percentile tra p0,0 e p99,9
`TotalTime`	Il numero di millisecondi durante i quali la richiesta è stata in transito dalla prospettiva del server. Questo valore viene misurato dal momento in cui MediaStore riceve la richiesta al momento in cui invia l'ultimo byte della risposta. Questo valore viene misurato dalla prospettiva del server perché misurazioni effettuate dalla prospettiva del client non sono influenzate dalla latenza di rete. Unità: millisecondi Dimensioni valide: • Nome del container • Nome del gruppo di oggetti • Tipo richiesta Statistiche valide: media, minimo (come P0,0), massimo (come p100), qualsiasi percentile tra p0,0 e p100

Metrica	Description
`TurnaroundTime`	Il numero di millisecondi MediaStore impiegato per l'elaborazione della richiesta. Questo valore viene misurato dal momento in cui MediaStore riceve l'ultimo byte della richiesta al momento in cui invia il primo byte della risposta. Unità: millisecondi Dimensioni valide: • Nome del container • Nome del gruppo di oggetti • Tipo richiesta Statistiche valide: media, minimo (come P0,0), massimo (come p100), qualsiasi percentile tra p0,0 e p100
`ThrottleCount`	Il numero di richieste HTTP inviate a MediaStore tale scopo è stato limitato. Unità: numero Dimensioni valide: • Nome del container • Nome del gruppo di oggetti • Tipo richiesta Statistiche valide: somma

Etichettatura delle risorse AWS MediaStore Elemental

Un tag è un'etichetta di attributo personalizzata che assegni o che AWS assegni a una risorsa. AWS Ogni tag è costituito da due parti:

- Una chiave del tag (ad esempio, `CostCenter`, `Environment` o `Project`). Le chiavi dei tag prevedono una distinzione tra lettere maiuscole e minuscole.

- Un campo facoltativo noto come valore del tag (ad esempio, `111122223333` o `Production`). Non specificare il valore del tag equivale a utilizzare una stringa vuota. Come le chiavi dei tag, i valori dei tag distinguono tra maiuscole e minuscole.

I tag consentono di:

- Identifica e organizza AWS le tue risorse. Molti servizi AWS supportano il tagging, perciò è possibile assegnare lo stesso tag a risorse di diversi servizi per indicare che le risorse sono correlate. Ad esempio, puoi assegnare a un AWS MediaStore *container* Elemental lo stesso tag che assegni a un input. AWS Elemental MediaLive
- Tenere traccia dei costi AWS. Puoi attivare questi tag sulla dashboard. Gestione dei costi e fatturazione AWS AWS utilizza i tag per organizzare in categorie i costi e invia all'utente un report mensile di allocazione dei costi. Per ulteriori informazioni, consulta la pagina sull'utilizzo dei tag per l'allocazione dei costi nella Guida per l'utente di AWS Billing.

Le seguenti sezioni forniscono ulteriori informazioni sui tag per AWS Elemental MediaStore.

Risorse supportate in AWS Elemental MediaStore

Le seguenti risorse in AWS Elemental MediaStore support tagging:

- *container*

Per informazioni sull'aggiunta e la gestione dei tag, consulta Gestione dei tag.

AWS Elemental MediaStore non supporta la funzionalità di controllo degli accessi basata su tag di AWS Identity and Access Management (IAM).

Convenzioni di denominazione e utilizzo dei tag

Le seguenti convenzioni di base di denominazione e utilizzo si applicano all'uso dei tag con le risorse AWS MediaStore Elemental:

- Ogni risorsa può avere un massimo di 50 tag.
- Per ciascuna risorsa, ogni chiave del tag deve essere univoca e ogni chiave del tag può avere un solo valore.
- La lunghezza massima delle chiavi di tag è 128 caratteri Unicode in UTF-8.

- Il valore massimo dei tag è 256 caratteri Unicode in UTF-8.
- I caratteri consentiti sono lettere, numeri, spazi rappresentabili in formato UTF-8, oltre ai seguenti caratteri: . : + = @ _ / - (trattino). EC2 Le risorse di Amazon consentono qualsiasi carattere.
- I valori e le chiavi dei tag rispettano la distinzione tra maiuscole e minuscole. Come best practice, è consigliabile definire una strategia per l'uso delle lettere maiuscole e minuscole nei tag e implementarla costantemente in tutti i tipi di risorse. Ad esempio, puoi decidere se utilizzare `Costcenter`, `costcenter` o `CostCenter` e utilizzare la stessa convenzione per tutti i tag. Non utilizzare tag simili con lettere maiuscole o minuscole incoerenti.
- Il `aws:` prefisso è vietato per i tag; è riservato all' AWS uso. Non è possibile modificare né eliminare le chiavi o i valori di tag con tale prefisso. I tag con questo prefisso non vengono conteggiati per la quota di tag per risorsa.

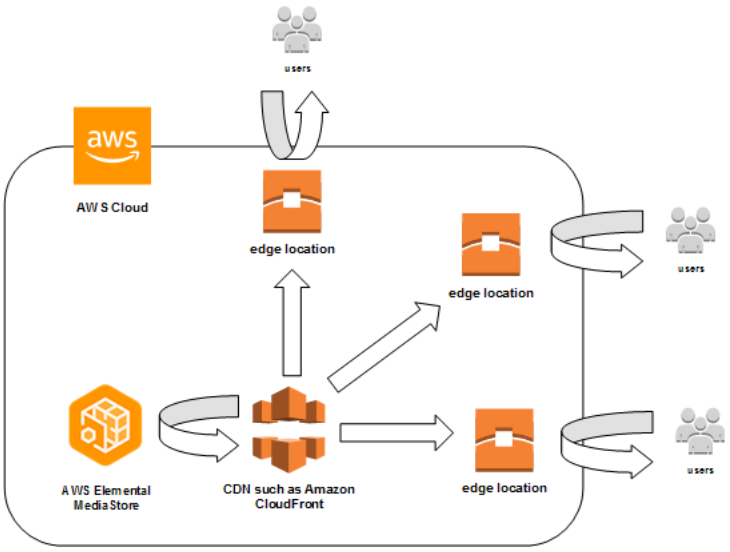
Gestione dei tag

I tag sono formati dalle proprietà `Key` e `Value` in una risorsa. È possibile utilizzare l'API AWS CLI o l' MediaStore API per aggiungere, modificare o eliminare i valori di queste proprietà. Per informazioni sull'utilizzo dei tag, consulta le seguenti sezioni nell'AWS Elemental MediaStore API Reference:

- CreateContainer
- ListTagsForResource
- Risorse
- TagResource
- UntagResource

Utilizzo di reti per la distribuzione di contenuti (CDNs)

Puoi utilizzare una rete di distribuzione di contenuti (CDN) come Amazon CloudFront per fornire i contenuti archiviati in MediaStore AWS Elemental. Una CDN è un insieme di server distribuiti a livello globale che effettua il caching di contenuti quali i video. Quando un utente richiede i tuoi contenuti, la CDN instrada la richiesta alla edge location che offre la latenza minore. Se il caching è già stato effettuato in tale edge location, la CDN distribuisce immediatamente i contenuti. Se i tuoi contenuti non si trovano attualmente in quella edge location, il CDN li recupera dall'origine (ad esempio dal MediaStore contenitore) e li distribuisce all'utente.



Argomenti

- Consentire CloudFront ad Amazon di accedere al tuo container AWS Elemental MediaStore
- Interazione di AWS MediaStore Elemental con le cache HTTP

Consentire CloudFront ad Amazon di accedere al tuo container AWS Elemental MediaStore

Puoi usare Amazon CloudFront per servire i contenuti archiviati in un contenitore in AWS Elemental MediaStore. Puoi farlo in uno dei seguenti modi:

- Utilizzo di Origin Access Control (OAC)- (Consigliato) Utilizzate questa opzione se Regione AWS supportate la funzionalità OAC di CloudFront.

- Usare Shared Secrets- Utilizzate questa opzione se Regione AWS non supportate la funzionalità OAC di. CloudFront

Utilizzo di Origin Access Control (OAC)

Puoi utilizzare la funzionalità Origin Access Control (OAC) di Amazon CloudFront per proteggere le origini AWS MediaStore Elemental con una maggiore sicurezza. Puoi abilitare AWS Signature Version 4 (SigV4) sulle CloudFront richieste di MediaStore origine e impostare quando e CloudFront se firmare le richieste. Puoi accedere alla funzionalità OAC CloudFront tramite la console APIs, l'SDK o la CLI e non sono previsti costi aggiuntivi per il suo utilizzo.

Per ulteriori informazioni sull'utilizzo della funzionalità OAC con MediaStore, consulta Limitazione dell'accesso a un' MediaStore origine nella Amazon CloudFront Developer Guide.

Usare Shared Secrets

Se Regione AWS non supporti la funzionalità OAC di Amazon CloudFront, puoi allegare una policy al tuo container AWS MediaStore Elemental che garantisca l'accesso in lettura o superiore a. CloudFront

> **Note**
>
> Ti consigliamo di utilizzare la funzionalità OAC se la supporti. Regione AWS Le seguenti procedure richiedono la configurazione MediaStore e l'utilizzo CloudFront di segreti condivisi per limitare l'accesso ai MediaStore contenitori. Per seguire le migliori pratiche di sicurezza, questa configurazione manuale richiede la rotazione periodica dei segreti. Con OAC on origin, MediaStore puoi indicare di firmare le richieste utilizzando SigV4 e inoltrarle CloudFront a MediaStore per la corrispondenza delle firme, eliminando la necessità di utilizzare e ruotare i segreti. Ciò garantisce che le richieste vengano verificate automaticamente prima della distribuzione dei contenuti multimediali, rendendo la consegna dei contenuti multimediali più semplice MediaStore e CloudFront sicura.

Per consentire l'accesso CloudFront al contenitore (console)

1. Apri la MediaStore console all'indirizzo https://console.aws.amazon.com/mediastore/.
2. Nella pagina Containers (Container), scegliere il nome del container.

Viene visualizzata la pagina dei dettagli del container.

3. Nella sezione Container policy, allega una policy che garantisca l'accesso in lettura o superiore ad Amazon CloudFront.

   Example

   La seguente politica di esempio, simile alla politica di esempio per l'accesso pubblico alla lettura tramite HTTPS, soddisfa questi requisiti perché consente `GetObject` e `DescribeObject` comanda chiunque invii richieste al tuo dominio tramite HTTPS. Inoltre, la seguente politica di esempio protegge meglio il flusso di lavoro perché consente CloudFront l'accesso agli MediaStore oggetti solo quando la richiesta avviene tramite una connessione HTTPS e contiene l'intestazione Referer corretta.

4. Nella sezione Container CORS policy (Policy CORS container), assegnare una policy che garantisca il livello di accesso desiderato.

   > **Note**
   >
   > Una policy CORS è necessaria solo per fornire l'accesso a un lettore basato su browser.

5. Annotare i dettagli riportati di seguito:

   - L'endpoint dati assegnato al tuo container . Questa informazione è reperibile nella sezione Info della pagina Containers (Container). Nel CloudFront, l'endpoint dei dati viene chiamato nome di dominio di origine.
   - La struttura della cartella nel container in cui gli oggetti vengono archiviati. In CloudFront, questo è indicato come percorso di origine. Questa impostazione è facoltativa. Per ulteriori informazioni sui percorsi di origine, consulta l'Amazon CloudFront Developer Guide.

6. Nel CloudFront, crea una distribuzione configurata per fornire contenuti da AWS Elemental MediaStore. Saranno necessarie le informazioni raccolte nella fase precedente.

Dopo aver associato la policy ai MediaStore contenitori, devi configurare CloudFront l'utilizzo solo delle connessioni HTTPS per le richieste di origine e aggiungere anche un'intestazione personalizzata con il valore segreto corretto.

Per configurare l'accesso CloudFront al contenitore tramite una connessione HTTPS con un valore segreto per l'intestazione Referer (console)

1. Apri la console. CloudFront
2. Nella pagina Origins, scegli la tua MediaStore origine.
3. Scegli Modifica.
4. Scegli HTTPS solo per il protocollo.
5. Nella sezione Aggiungi intestazione personalizzata, scegli Aggiungi intestazione.
6. Per il nome, scegli Referer. Per il valore, usa la stessa *<secretValue>* stringa che hai usato nella politica del contenitore.
7. Scegli Salva e lascia che le modifiche vengano distribuite.

Interazione di AWS MediaStore Elemental con le cache HTTP

AWS Elemental MediaStore archivia gli oggetti in modo che possano essere memorizzati nella cache in modo corretto ed efficiente da reti di distribuzione di contenuti () CDNs come Amazon. CloudFront Quando un utente finale o un CDN recupera un oggetto da MediaStore, il servizio restituisce intestazioni HTTP che influiscono sul comportamento di memorizzazione nella cache dell'oggetto. (Gli standard per il comportamento di memorizzazione nella cache HTTP 1.1 si trovano nella sezione 13.) RFC2616 Queste intestazioni sono:

- **`ETag`** (non personalizzabile): l'intestazione del tag entità è un identificatore univoco per la risposta inviata da MediaStore . I browser Web CDNs e conformi agli standard utilizzano questo tag come chiave per memorizzare nella cache l'oggetto. MediaStore genera automaticamente un file `ETag` per ogni oggetto quando viene caricato. È possibile visualizzare i dettagli di un oggetto per determinarne il ETag valore.
- **`Last-Modified`**(non personalizzabile): il valore di questa intestazione indica la data e l'ora in cui l'oggetto è stato modificato. MediaStore genera automaticamente questo valore quando l'oggetto viene caricato.
- **`Cache-Control`** (personalizzabile): il valore di questa intestazione controlla per quanto tempo un oggetto deve essere memorizzato nella cache prima che la CDN controlli se è stato modificato. Puoi impostare questa intestazione su qualsiasi valore quando carichi un oggetto in un MediaStore contenitore utilizzando la CLI o l'API. Il set completo dei valori validi è descritto nella documentazione HTTP/1.1. Se non imposti questo valore quando carichi un oggetto, MediaStore non restituirà questa intestazione quando l'oggetto viene recuperato.

Un caso di utilizzo comune per l'intestazione Cache-Control consiste nel specificare una durata per la memorizzazione dell'oggetto nella cache. Supponi, ad esempio, di avere un file manifest video che viene spesso sovrascritto da un codificatore. Puoi impostare `max-age` su 10 per indicare che l'oggetto deve essere memorizzato nella cache per soli 10 secondi. In alternativa supponi di avere un segmento video memorizzato che non verrà mai sovrascritto. Puoi impostare `max-age` per questo oggetto su 31536000 per memorizzare l'oggetto nella cache per circa 1 anno.

Richieste condizionali

Richieste condizionali a MediaStore

MediaStore risponde in modo identico alle richieste condizionali (utilizzando intestazioni di richiesta come `If-Modified-Since` e`If-None-Match`, come descritto in RFC7232) e alle richieste incondizionate. Ciò significa che quando MediaStore riceve una `GetObject` richiesta valida, il servizio restituisce sempre l'oggetto anche se il client ha già l'oggetto.

Richieste condizionali a CDNs

CDNs che forniscono contenuti per conto di MediaStore possono elaborare le richieste condizionali restituendole`304 Not Modified`, come descritto nella RFC7232 sezione 4.1. Ciò significa che non è necessario trasferire il contenuto completo dell'oggetto, poiché il richiedente dispone già di un oggetto che corrisponde alla richiesta condizionale.

CDNs (e altre cache conformi a HTTP/1.1) basano queste decisioni sulle `Cache-Control` intestazioni `ETag` and inoltrate dai server di origine. Per controllare la frequenza con cui CDNs interroga i server di MediaStore origine per gli aggiornamenti degli oggetti recuperati ripetutamente, imposta le `Cache-Control` intestazioni per tali oggetti quando li carichi. MediaStore

Utilizzo di questo servizio con un SDK AWS

AWS i kit di sviluppo software (SDKs) sono disponibili per molti linguaggi di programmazione più diffusi. Ogni SDK fornisce un'API, esempi di codice, e documentazione che facilitano agli sviluppatori la creazione di applicazioni nel loro linguaggio preferito.

Documentazione sugli SDK	Esempi di codice
AWS SDK per C++	AWS SDK per C++ esempi di codice
AWS CLI	AWS CLI esempi di codice
AWS SDK per Go	AWS SDK per Go esempi di codice
AWS SDK per Java	AWS SDK per Java esempi di codice
AWS SDK per JavaScript	AWS SDK per JavaScript esempi di codice
AWS SDK per Kotlin	AWS SDK per Kotlin esempi di codice
AWS SDK per .NET	AWS SDK per .NET esempi di codice
AWS SDK per PHP	AWS SDK per PHP esempi di codice
AWS Strumenti per PowerShell	AWS Strumenti per PowerShell esempi di codice
AWS SDK per Python (Boto3)	AWS SDK per Python (Boto3) esempi di codice
AWS SDK per Ruby	AWS SDK per Ruby esempi di codice
AWS SDK per Rust	AWS SDK per Rust esempi di codice
AWS SDK per SAP ABAP	AWS SDK per SAP ABAP esempi di codice
AWS SDK per Swift	AWS SDK per Swift esempi di codice

Per esempi specifici del servizio, consulta Esempi di codice per MediaStore l'utilizzo AWS SDKs.

Esempio di disponibilità

Non riesci a trovare quello che ti serve? Richiedi un esempio di codice utilizzando il link Provide feedback (Fornisci un feedback) nella parte inferiore di questa pagina.

Esempi di codice per MediaStore l'utilizzo AWS SDKs

I seguenti esempi di codice mostrano come utilizzare un kit MediaStore di sviluppo AWS software (SDK).

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice

- Esempi di base per l' MediaStore utilizzo AWS SDKs
  - Azioni per l'utilizzo MediaStore AWS SDKs
    - Utilizzo CreateContainer con un AWS SDK o una CLI
    - Utilizzo DeleteContainer con un AWS SDK o una CLI
    - Utilizzare DeleteObject con un SDK AWS
    - Utilizzo DescribeContainer con un AWS SDK o una CLI
    - Utilizzo GetObject con un AWS SDK o una CLI
    - Utilizzo ListContainers con un AWS SDK o una CLI
    - Utilizzo PutObject con un AWS SDK o una CLI

Esempi di base per l' MediaStore utilizzo AWS SDKs

I seguenti esempi di codice mostrano come utilizzare le nozioni di base di AWS Elemental MediaStore with. AWS SDKs

Esempi

- Azioni per l'utilizzo MediaStore AWS SDKs
  - Utilizzo CreateContainer con un AWS SDK o una CLI
  - Utilizzo DeleteContainer con un AWS SDK o una CLI
  - Utilizzare DeleteObject con un SDK AWS

- Utilizzo DescribeContainer con un AWS SDK o una CLI
- Utilizzo GetObject con un AWS SDK o una CLI
- Utilizzo ListContainers con un AWS SDK o una CLI
- Utilizzo PutObject con un AWS SDK o una CLI

Azioni per l'utilizzo MediaStore AWS SDKs

I seguenti esempi di codice mostrano come eseguire singole MediaStore azioni con AWS SDKs. Ogni esempio include un collegamento a GitHub, dove sono disponibili le istruzioni per la configurazione e l'esecuzione del codice.

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per un elenco completo, consulta la Documentazione di riferimento delle API AWS Elemental MediaStore.

Esempi

- Utilizzo CreateContainer con un AWS SDK o una CLI
- Utilizzo DeleteContainer con un AWS SDK o una CLI
- Utilizzare DeleteObject con un SDK AWS
- Utilizzo DescribeContainer con un AWS SDK o una CLI
- Utilizzo GetObject con un AWS SDK o una CLI
- Utilizzo ListContainers con un AWS SDK o una CLI
- Utilizzo PutObject con un AWS SDK o una CLI

Utilizzo **`CreateContainer`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `CreateContainer`.

CLI

AWS CLI

Per creare un contenitore

L'`create-container`esempio seguente crea un nuovo contenitore vuoto.

```
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"
    }
}
```

Per ulteriori informazioni, consulta Creazione di un contenitore nella Guida MediaStore utente di AWS Elemental.

- Per i dettagli sull'API, consulta CreateContainer AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

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

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

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

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

                Usage:    <containerName>

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

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

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

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


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

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

```
                Thread.sleep(sleepTime * 1000);
            }

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

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

- Per i dettagli sull'API, consulta la CreateContainersezione AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteContainer`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `DeleteContainer`.

CLI

AWS CLI

Per eliminare un contenitore

L'`delete-container`esempio seguente elimina il contenitore specificato. Puoi eliminare un container solo se non contiene oggetti.

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

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Eliminazione di un contenitore nella Guida utente di AWS MediaStore Elemental.

- Per i dettagli sull'API, consulta AWS CLI Command DeleteContainerReference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

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

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

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

                Usage:    <containerName>

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

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

```
        }

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

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


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

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

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

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

- Per i dettagli sull'API, consulta la DeleteContainersezione AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`DeleteObject`** con un SDK AWS

Il seguente esempio di codice mostra come utilizzare`DeleteObject`.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

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

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

```
 */
public class DeleteObject {
    public static void main(String[] args) throws URISyntaxException {
        final String usage = """

                Usage:    <completePath> <containerName>

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

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

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

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

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

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

            mediaStoreData.deleteObject(deleteObjectRequest);

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

```
            System.exit(1);
        }
    }

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

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

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

- Per i dettagli sull'API, consulta la DeleteObjectsezione AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DescribeContainer`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `DescribeContainer`.

CLI

AWS CLI

Per visualizzare i dettagli di un contenitore

L'`describe-container`esempio seguente visualizza i dettagli del contenitore specificato.

```
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"
    }
}
```

Per ulteriori informazioni, consulta Visualizzazione dei dettagli di un contenitore nella Guida per l' MediaStore utente di AWS Elemental.

- Per i dettagli sull'API, consulta DescribeContainer AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

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

/**
 * Before running this Java V2 code example, set up your development
```

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

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

                Usage:    <containerName>

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

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

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

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

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

            DescribeContainerResponse containerResponse =
 mediaStoreClient.describeContainer(describeContainerRequest);
```

```
            System.out.println("The container name is " +
 containerResponse.container().name());
            System.out.println("The container ARN is " +
 containerResponse.container().arn());
            return containerResponse.container().status().toString();

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

- Per i dettagli sull'API, consulta la DescribeContainersezione AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetObject`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `GetObject`.

CLI

AWS CLI

Per scaricare un oggetto

L'`get-object`esempio seguente scarica un oggetto nell'endpoint specificato.

```
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
}
```

Per scaricare parte di un oggetto

L'`get-object`esempio seguente scarica una parte di un oggetto nell'endpoint specificato.

```
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"
}
```

Per ulteriori informazioni, consultate Downloading an Object nella Guida per l' MediaStore utente di AWS Elemental.

- Per i dettagli sull'API, consulta GetObject AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.core.ResponseInputStream;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.mediastore.MediaStoreClient;
import software.amazon.awssdk.services.mediastore.model.DescribeContainerRequest;
import
 software.amazon.awssdk.services.mediastore.model.DescribeContainerResponse;
import software.amazon.awssdk.services.mediastoredata.MediaStoreDataClient;
import software.amazon.awssdk.services.mediastoredata.model.GetObjectRequest;
import software.amazon.awssdk.services.mediastoredata.model.GetObjectResponse;
import
 software.amazon.awssdk.services.mediastoredata.model.MediaStoreDataException;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.net.URI;
import java.net.URISyntaxException;

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

                Usage:    <completePath> <containerName> <savePath>

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

        if (args.length != 3) {
            System.out.println(usage);
```

```
            System.exit(1);
        }

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

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

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

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

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

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

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

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

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

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

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

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

- Per i dettagli sull'API, consulta la GetObjectsezione AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListContainers`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `ListContainers`.

CLI

AWS CLI

Per visualizzare un elenco di contenitori

L'`list-containers`esempio seguente visualizza un elenco di tutti i contenitori associati all'account.

```
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"
        }
    ]
}
```

Per ulteriori informazioni, consulta Visualizzazione di un elenco di contenitori nella Guida per l' MediaStore utente di AWS Elemental.

- Per i dettagli sull'API, consulta ListContainers AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.mediastore.MediaStoreClient;
import software.amazon.awssdk.services.mediastore.model.Container;
```

```
import software.amazon.awssdk.services.mediastore.model.ListContainersResponse;
import software.amazon.awssdk.services.mediastore.model.MediaStoreException;
import java.util.List;

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

    public static void main(String[] args) {

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

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

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

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

- Per i dettagli sull'API, consulta la ListContainerssezione AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutObject`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `PutObject`.

CLI

AWS CLI

Per caricare un oggetto

L'`put-object`esempio seguente carica un oggetto nel contenitore specificato. È possibile specificare il percorso della cartella in cui salvare l'oggetto all'interno del contenitore. Se la cartella esiste già, AWS Elemental MediaStore memorizza l'oggetto nella cartella. Se la cartella non esiste, il servizio la crea e quindi memorizza l'oggetto nella cartella.

```
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"
}
```

Per ulteriori informazioni, consulta Caricamento di un oggetto nella Guida MediaStore utente di AWS Elemental.

- Per i dettagli sull'API, consulta AWS CLI Command PutObjectReference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

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

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

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

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

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

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

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

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

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

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

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

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

```
    }

    public static String getEndpoint(String containerName) {

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

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

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

- Per i dettagli sull'API, consulta la PutObjectsezione AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Quote in AWS Elemental MediaStore

La console Service Quotas fornisce informazioni sulle quote AWS MediaStore Elemental. Oltre a visualizzare le quote predefinite, è possibile utilizzare la console Service Quotas per richiedere aumenti di quota per le quote modificabili.

La tabella seguente descrive le quote, precedentemente denominate limiti, in AWS Elemental. MediaStore Le quote sono il numero massimo di risorse o operazioni di servizio per il tuo account AWS.

> **Note**
>
> Per assegnare quote a singoli contenitori all'interno del tuo account, contatta AWS Support o il tuo account manager. Questa opzione può aiutarti a suddividere i limiti a livello di account tra i contenitori, per evitare che un contenitore utilizzi l'intera quota.

Operazione o risorsa	Quota predefinita	Commenti
Container	100	Numero massimo di container che puoi creare in questo account.
Livelli di cartella	10	Numero massimo di livelli di cartella che puoi creare in un container. Puoi creare il numero di cartelle che desideri, per non più di 10 livelli all'interno di un container.
Cartelle	Illimitato	Puoi creare il numero di cartelle che desideri, per non più di 10 livelli all'interno di un container.
Dimensione oggetto	25 MB	Dimensione massima del file di un singolo oggetto.
Oggetti	Illimitato	Puoi caricare tutti gli oggetti che desideri in una cartella o in un contenitore nel tuo account.

Operazione o risorsa	Quota predefinita	Commenti
Frequenza delle richieste API DeleteObject	100	Il numero massimo di richieste di operazioni che puoi effettuare al secondo. Ulteriori richieste verranno sottoposte a limitazione (della larghezza di banda della rete). È possibile richiedere un aumento della quota.
Frequenza delle richieste API DescribeObject	1.000	Il numero massimo di richieste di operazioni che puoi effettuare al secondo. Ulteriori richieste verranno sottoposte a limitazione (della larghezza di banda della rete). È possibile richiedere un aumento della quota.
Frequenza delle richieste GetObject API per la disponibilità dei caricamenti standard	1.000	Il numero massimo di richieste di operazioni che puoi effettuare al secondo. Ulteriori richieste verranno sottoposte a limitazione (della larghezza di banda della rete). È possibile richiedere un aumento della quota.
Frequenza delle richieste GetObject API per la disponibilità dei caricamenti in streaming	25	Il numero massimo di richieste di operazioni che puoi effettuare al secondo. Ulteriori richieste verranno sottoposte a limitazione (della larghezza di banda della rete). È possibile richiedere un aumento della quota.
Frequenza delle richieste API ListItems	5	Il numero massimo di richieste di operazioni che puoi effettuare al secondo. Ulteriori richieste verranno sottoposte a limitazione (della larghezza di banda della rete). È possibile richiedere un aumento della quota.

Operazione o risorsa	Quota predefinita	Commenti
Frequenza delle richieste PutObject API per la codifica di trasferimento in blocchi (nota anche come disponibilità di upload in streaming)	10	Il numero massimo di richieste di operazioni che puoi effettuare al secondo. Ulteriori richieste verranno sottoposte a limitazione (della larghezza di banda della rete). È possibile richiedere un aumento della quota. Nella richiesta, specificare il TPS richiesto e la dimensione media dell'oggetto.
Frequenza delle richieste PutObject API per la disponibilità di caricamento standard	100	Il numero massimo di richieste di operazioni che puoi effettuare al secondo. Ulteriori richieste verranno sottoposte a limitazione (della larghezza di banda della rete). È possibile richiedere un aumento della quota. Nella richiesta, specificare il TPS richiesto e la dimensione media dell'oggetto.
Regole di una policy di parametro	10	Numero massimo di regole che è possibile includere in una policy di parametro.
Regole in una policy del ciclo di vita degli oggetti	10	Il numero massimo di regole che puoi includere in una policy del ciclo di vita degli oggetti.

Informazioni relative a AWS Elemental MediaStore

La tabella seguente elenca le risorse correlate che troverai utili quando lavori con AWS Elemental MediaStore.

- Corsi e workshop: collegamenti a corsi specializzati e basati su ruoli, oltre a laboratori di autoapprendimento per aiutarti ad affinare le tue abilità e acquisire esperienza pratica. AWS
- AWS Developer Center: esplora i tutorial, scarica strumenti e scopri gli eventi per sviluppatori. AWS
- AWS Strumenti per sviluppatori: collegamenti a strumenti di sviluppo SDKs, toolkit IDE e strumenti da riga di comando per lo sviluppo e la gestione di applicazioni. AWS
- Centro risorse introduttivo: scopri come configurare Account AWS, entrare a far parte della AWS community e lanciare la tua prima applicazione.
- Tutorial pratici: segui i tutorial per avviare la step-by-step tua prima applicazione su. AWS
- AWS Whitepaper: collegamenti a un elenco completo di AWS white paper tecnici, su argomenti quali architettura, sicurezza ed economia e redatti da Solutions Architects o altri esperti tecnici. AWS
- Supporto AWS Center: l'hub per la creazione e la gestione dei casi. Supporto AWS Include anche collegamenti ad altre risorse utili, come forum, informazioni tecniche FAQs, stato di salute del servizio e AWS Trusted Advisor.
- Supporto— La pagina web principale per informazioni su Supporto one-on-one, un canale di supporto a risposta rapida per aiutarti a creare ed eseguire applicazioni nel cloud.
- Contatti - Un punto di contatto centrale per richieste relative a fatturazione, account, eventi, uso illecito e altre questioni relative ad AWS .
- AWS Termini del sito: informazioni dettagliate sul nostro copyright e marchio, sull'account, sulla licenza e sull'accesso al sito e altri argomenti.

Cronologia dei documenti per la Guida per l'utente

La tabella seguente descrive la documentazione per questa versione di AWS Elemental MediaStore. Per ricevere notifiche sugli aggiornamenti di questa documentazione, è possibile sottoscrivere un feed RSS.

Modifica	Descrizione	Data
Avviso di fine del supporto	Avviso di fine del supporto: il 13 novembre 2025 AWS interromperà il supporto per AWS Elemental. MediaStor e Dopo il 13 novembre 2025, non potrai più accedere alla console o alle MediaStor e risorse. MediaStore Per ulteriori informazioni, consulta questo post del blog.	12 novembre 2024
Miglioramento di Origin Access Control (OAC)	Sono state aggiunte informazi oni su come utilizzare OAC con MediaStore AWS Elemental.	17 aprile 2023
Aggiornamenti delle quote	Valore e descrizione della quota corretti per. `Rules in a Metric Policy`	25 ottobre 2022
ExpiresAt campo	I log di accesso ora includono un `ExpiresAt` campo che indica la data e l'ora di scadenza dell'oggetto in base alle regole transitorie sui dati contenute nella politica del ciclo di vita del contenitore.	16 luglio 2020

Regole di transizione del ciclo di vita	Puoi ora aggiungere una regola di transizione del ciclo di vita alla policy del ciclo di vita dell'oggetto che imposta gli oggetti da spostare nella classe di archiviazione con accesso non frequente (IA) dopo aver raggiunto una certa età.	20 aprile 2020
Contenitore vuoto	Puoi eliminare tutti gli oggetti all'interno di un container contemporaneamente.	7 aprile 2020
Support per i CloudWatch parametri di Amazon	Puoi impostare una politica dei parametri per stabilire a quali MediaStore metriche inviare. CloudWatch	30 marzo 2020
Wildcard nelle regole di eliminazione degli oggetti	In una policy del ciclo di vita degli oggetti, è ora possibile utilizzare un carattere jolly in una regola dell'oggetto di eliminazione. Ciò consente di specificare i file in base al nome del file o all'estensione che si desidera eliminare dal servizio dopo un certo numero di giorni.	20 dicembre 2019
Politiche relative al ciclo di vita degli oggetti	Ora puoi aggiungere una regola alla policy del ciclo di vita degli oggetti che indica la scadenza per età in secondi.	13 settembre 2019

CloudFormation supporto	Ora puoi usare un CloudFormation modello per creare automaticamente un contenitore. Il modello CloudFormation gestisce i dati per cinque operazioni API: creazione di un container, impostazione della registrazione degli accessi, aggiornamento della policy del container di default, aggiunta di una policy CORS e aggiunta della policy del ciclo di vita degli oggetti.	17 maggio 2019
Quote per la disponibilità dei caricamenti in streaming	Per gli oggetti con disponibilità di upload in streaming (trasferimento a pezzi di oggetti), l'operazione `PutObject` non può superare i 10 TPS e l'operazione `GetObject` non può superare i 25 TPS.	8 Aprile 2019
Trasferimento in blocchi di oggetti	Aggiunto il supporto per il trasferimento a blocchi di oggetti. Questa funzionalità consente di specificare che un oggetto è disponibile per il download prima che sia completamente caricato.	5 aprile 2019
Registrazione degli accessi	AWS Elemental MediaStore ora supporta la registrazione degli accessi, che fornisce record dettagliati per le richieste effettuate agli oggetti in un contenitore.	25 febbraio 2019

Politiche relative al ciclo di vita degli oggetti	Aggiunto il supporto per le policy del ciclo di vita degli oggetti, che gestiscono la data di scadenza di oggetti all'interno del container corrente.	12 dicembre 2018
Quota di dimensioni degli oggetti aumentata	La quota della dimensione di un oggetto è ora di 25 MB.	10 ottobre 2018
Quota di dimensioni degli oggetti aumentata	La quota per la dimensione di un oggetto è ora di 20 MB.	6 settembre 2018
AWS CloudTrail integration	Il contenuto dell' CloudTrail integrazione è stato aggiornato per allinearlo alle recenti modifiche al CloudTrail servizio.	12 luglio 2018
Collaborazione CDN	Sono state aggiunte informazioni su come utilizzare AWS Elemental MediaStore con una rete di distribuzione di contenuti (CDN) come Amazon. CloudFront	14 aprile 2018
Configurazioni CORS	AWS Elemental MediaStore ora supporta la condivisione di risorse tra origini diverse (CORS), che consente alle applicazioni Web client caricate in un dominio di interagire con le risorse di un dominio diverso.	7 febbraio 2018

Nuovo servizio e guida	Questa è la versione iniziale del servizio di origine e archiviazione video AWS Elemental e della AWS MediaStore Elemental User Guide. MediaStore	27 Novembre 2017

Note

- I servizi AWS multimediali non sono progettati o destinati all'uso con applicazioni o in situazioni che richiedono prestazioni sicure, come operazioni di sicurezza, sistemi di navigazione o comunicazione, controllo del traffico aereo o macchine di supporto vitale in cui l'indisponibilità, l'interruzione o il fallimento dei servizi potrebbero causare morte, lesioni personali, danni materiali o danni ambientali.

AWS Glossario

Per la AWS terminologia più recente, consultate il AWS glossario nella sezione Reference.Glossario AWS