aws

Guida per l'utente

# AWSElementale MediaStore

# AWSElementale MediaStore: Guida per l'utente

# Table of Contents

# 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 live. 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.

Percorso

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 globale di rete per la distribuzione di contenuti (CDN) 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.
- AWS CloudFormation è un servizio che consente di modellare e configurare le risorse AWS. Crei un modello che descrive tutte le AWS risorse che desideri (come i MediaStore contenitori) e AWS CloudFormation si occupa del provisioning e della configurazione di tali risorse per te. Non è necessario creare e configurare singolarmente le risorse AWS e determinare le dipendenze, perché è AWS CloudFormation a gestire tutti questi aspetti. Per ulteriori informazioni, consulta la Guida per l'utente AWS CloudFormation.
- 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 Guida per l'utente AWS CloudTrail.
- Amazon CloudWatch è un servizio di monitoraggio delle risorse AWS cloud e delle applicazioni su cui eseguiAWS. 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 aiuta a controllare in modo sicuro l'accesso alle risorse AWS per gli utenti. Utilizza IAM per stabilire chi può utilizzare le tue risorse (autenticazione) AWS, quali risorse e in che modo (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 MediaStoreAPI 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
```

- SDK AWS: se usi un linguaggio di programmazione per cui AWS fornisce un SDK, puoi utilizzare un SDK per accedere a MediaStore. Gli SDK semplificano l'autenticazione, si integrano senza difficoltà nel tuo ambiente di sviluppo e ti offrono semplice accesso ai comandi di MediaStore . Per ulteriori informazioni, consulta Strumenti per Amazon Web Services.
- AWS Tools per Windows PowerShell: per ulteriori informazioni, consulta la Guida per AWS Tools for Windows 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 a 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 passaggi seguenti.

Argomenti



## Registrati per un Account AWS

Se non disponi di un 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.

   Nel corso della procedura di registrazione riceverai una telefonata, durante la quale sarà necessario inserire un codice di verifica attraverso la tastiera del telefono.

   Quando ti iscrivi a un Account AWS, un Utente root dell'account AWSviene creato. L'utente root ha accesso a tutti Servizi AWS e le risorse presenti 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 per un Account AWS, proteggi il tuo Utente root dell'account AWS, abilita AWS IAM Identity Center 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 a AWS Management Consolecome proprietario dell'account selezionando Utente root e inserendo il Account AWS indirizzo email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso tramite utente root, consulta Accesso come utente root in Accedi ad AWS Guida per l'utente.
2. Attiva l'autenticazione a più fattori (MFA) per il tuo utente root.

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

Crea un utente con accesso amministrativo

1. Abilita IAM Identity Center.

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

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

Accesso come utente amministratore

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

  Per informazioni sull'accesso tramite un utente di IAM Identity Center, vedi Accesso a AWS accedere al portale in Accedi ad AWS Guida per l'utente.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni che segua la migliore pratica di applicazione delle autorizzazioni con privilegi minimi.

   Per istruzioni, vedere Creare un set di autorizzazioni nella AWS IAM Identity Center Guida per l'utente.
2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

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

# Nozioni base su AWS Elemental MediaStore

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

Argomenti



## Fase 1: Accedere ad AWS Elemental MediaStore

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

Per accedere ad AWS Elemental MediaStore

- Accedere aAWS Management Console e aprire 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 i 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 del percorso non esiste ancora, AWS Elemental le 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).

# Container in AWS ElementalMediaStore

Usa i container in MediaStore per archiviare le cartelle e gli 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, consultaAWS ElementalMediaStorePrezzi.

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 container automaticamente utilizzando un modello AWS CloudFormation. Il modello AWS 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, consultare la Guida per l'utente AWS CloudFormation.

Per creare un container (console)

1. Apertura dellaMediaStoreConsole inhttps://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, consultare . Regole per i nomi di container .
4. ScegliereCreazione di container. AWS ElementalMediaStoreaggiunge il nuovo container a un elenco di container. Inizialmente, lo stato del container è Creating (In fase di creazione), quindi diventa Active (Attivo).

Per creare un container (AWS CLI)

- In AWS CLI, usa il comando `create-container`.

```
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. Apertura dellaMediaStoreConsole inhttps://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 comando `describe-container`.

```
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)

- Apertura dellaMediaStoreConsole inhttps://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 comando `list-containers`.

  ```
  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. Apertura dellaMediaStoreConsole inhttps://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 comando `delete-container`.

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

  Il comando non ha un valore restituito.

# Policy di AWS ElementalMediaStore

Puoi applicare una o più di queste policy al AWS ElementalMediaStorecontainer:

- Policy di container- Imposta i diritti di accesso a tutte le cartelle e gli oggetti all'interno del container. MediaStoreimposta un criterio predefinito che consente agli utenti di eseguire tuttiMediaStoreoperazioni sul contenitore. Questa policy specifica che tutte le operazioni devono essere eseguite su HTTPS. Dopo aver creato un container, puoi modificarne la policy.
- Policy CORS (CRS-Origin Resource Sharing)- Consente alle applicazioni Web client caricate da un dominio di interagire con le risorse caricate in un dominio differente. MediaStorenon imposta un criterio CORS predefinito.
- Policy di parametro- ConsenteMediaStoreper inviare parametri ad AmazonCloudWatch. MediaStorenon imposta una policy di parametro di default.
- Policy del ciclo di vita degli oggetti- Controlla la durata di permanenza degli oggetti in unMediaStorecontainer. MediaStorenon imposta una policy del ciclo di vita degli oggetti di default.

## Politiche per i container in AWS ElementalMediaStore

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 di default, che viene automaticamente collegata a tutti i nuovi container, consente l'accesso a tutti i AWS ElementalMediaStoreoperazioni 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 la AWS CLI per visualizzare la policy basata su risorse di un container.

Per visualizzare una policy di container (console)

1. Apertura dellaMediaStoreConsole alhttps://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 comando `get-container-policy`.

```
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. Apertura dellaMediaStoreConsole alhttps://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:

```
{
  "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 comando `put-container-policy`.

```
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 (predefinito)
- 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 (predefinito)

Quando crei un contenitore, AWS ElementalMediaStoreapplica automaticamente le seguenti policy basate su risorse:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MediaStoreFullAccess",
      "Action": [ "mediastore:*" ],
      "Principal":{
          "AWS" : "arn:aws:iam::<aws_account_number>:root"},
      "Effect": "Allow",
```

```
      "Resource": "arn:aws:mediastore:<region>:<owner acct number>:container/<container
 name>/*",
      "Condition": {
        "Bool": { "aws:SecureTransport": "true" }
      }
    }
  ]
}
```

La policy è integrata nel servizio, quindi non è necessario crearla. Tuttavia, è possibilemodifica della policynel contenitore se le autorizzazioni nel criterio predefinito non sono allineate con le autorizzazioni che si desidera 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 accesso in lettura a chiunque su una connessione SSL/TLS protetta, utenti autenticati e anonimi (utenti che non sono connessi). 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:

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

```
  ]
}
```

## 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):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadOverHttpOrHttps",
      "Effect": "Allow",
      "Action": ["mediastore:GetObject", "mediastore:DescribeObject"],
      "Principal": "*",
      "Resource": "arn:aws:mediastore:<region>:<owner acct number>:container/<container
 name>/*",
      "Condition": {
        "Bool": { "aws:SecureTransport": ["true", "false"] }
      }
    }
  ]
}
```

## 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 in un server con un certificato SSL/TLS:

```
{
  "Version" : "2012-10-17",
  "Statement" : [ {
    "Sid" : "CrossAccountReadOverHttpOrHttps",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::<other acct number>:root"
    },
    "Action" : [ "mediastore:GetObject", "mediastore:DescribeObject" ],
```

```
    "Resource" : "arn:aws:mediastore:<region>:<owner acct number>:container/<container
 name>/*",
    "Condition" : {
      "Bool" : {
        "aws:SecureTransport" : [ "true", "false" ]
      }
    }
  } ]
}
```

## Policy di container di esempio: Accesso in lettura multiaccount su HTTPS

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

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

## 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.

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

## Policy di container di esempio: Accesso limitato a indirizzi IP specifici

Questa policy di esempio consente l'accesso a tutte le AWS ElementalMediaStoreoperazioni su oggetti nel container specificato. La richiesta deve, tuttavia, avere origine dall'intervallo di indirizzi IP specificati nella condizione.

La condizione in questa istruzione identifica l'intervallo 198.51.100.* di indirizzi IP di Internet Protocol versione 4 (IPv4) consentiti, con un'unica 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. I valori IPv4 `aws:sourceIp` utilizzano la notazione CIDR standard. Per ulteriori informazioni, consultaOperatori di condizione con indirizzo IPNella Guida per l'utente di IAM.

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

# Policy CORS (Cross-Origin Resource Sharing) in AWS ElementalMediaStore

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 ElementalMediaStore, è possibile creare applicazioni Web lato client complete conMediaStoree consenti in modo selettivo l'accesso cross-origin al tuoMediaStorerisorse AWS.

> **Note**
>
> Se utilizzi AmazonCloudFrontper distribuire contenuti da un contenitore che dispone di una politica CORS, assicurati diconfigurare la distribuzione per AWS ElementalMediaStore(incluso il passaggio per modificare il comportamento della cache per configurare CORS).

In questa sezione viene fornita una panoramica della funzionalità CORS. Negli argomenti secondari viene descritto come abilitare la funzionalità CORS utilizzando AWS ElementalMediaStoreconsole o utilizzando a livello di programmazioneMediaStoreAPI REST e gli SDK AWS.

Argomenti



## Scenari di casi d'uso di CORS

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

- Scenario 1: Supponi di distribuire video in streaming live in un AWS ElementalMediaStoredenominato containerLiveVideo. 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`. Intendi utilizzare unaJavaScriptper accedere a tutti i video provenienti da questo container tramite non autenticati`GET`e`PUT`. In genere, un browser bloccaJavaScriptdall'consentire queste richieste, ma puoi impostare una policy CORS nel container in modo da consentire esplicitamente queste richieste da`www.example.com`.

- Scenario 2: Supponi di voler ospitare la stessa diretta streaming dello Scenario 1 dalMediaStorecontainer, ma vogliono consentire 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

In questa sezione viene descritto come aggiungere una configurazione CORS (Cross-Origin Resource Sharing) a un AWS ElementalMediaStore. 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. Apertura dellaMediaStorela consolehttps://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 comando `put-cors-policy`.

```
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. Apertura dellaMediaStorela consolehttps://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 comando `get-cors-policy`.

```
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. Apertura dellaMediaStorela consolehttps://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 comando `put-cors-policy`.

```
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. Apertura dellaMediaStorela consolehttps://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 comando `delete-cors-policy`.

  ```
  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, consultare 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 l'intestazione non è presente, AWS ElementalMediaStorenon considera la richiesta come una richiesta multiorigine e non riinvia le intestazioni della 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
- Policy CORS di esempio: Accesso in lettura per un dominio specifico

### Policy CORS di esempio: Accesso in lettura per qualsiasi dominio

La policy seguente consente a una pagina Web da qualsiasi dominio di recuperare contenuti dal AWS ElementalMediaStore. 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 policy seguente consente a una pagina Web dahttps://www.example.comper recuperare contenuti dal tuo AWS ElementalMediaStore. La richiesta include tutte le intestazioni HTTP da https://www.example.come 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
  }
]
```

## Policy del ciclo di vita degli oggetti in AWS ElementalMediaStore

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 ElementalMediaStoreelimina gli oggetti. Puoi eliminare gli oggetti quando non sono più necessari per risparmiare sui costi di storage.

Puoi inoltre specificare che MediaStore deve spostare gli oggetti nella classe di archiviazione con accesso non frequente (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, consultare Prezzi di MediaStore.

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, consultare . 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 la durata di permanenza degli oggetti in un AWS ElementalMediaStorecontainer. 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 affinché MediaStore applichi la nuova policy al container.

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 transitori non emettono unCloudWatchevento quando scadono.

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. L'applicazione della nuova policy da parte di MediaStore richiede fino a 20 minuti, ma possono essere necessarie fino a 24 ore prima che gli oggetti vengano cancellati dal container.

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, consultare Prezzi di MediaStore.

Una volta che un oggetto si è spostato nella classe di storage IA, non è possibile spostarlo nella classe di storage 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. L'applicazione della nuova policy da parte di MediaStore richiede fino a 20 minuti, ma possono essere necessarie fino a 24 ore prima che gli oggetti vengano cancellati dal container.

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 AWS ElementalMediaStoreper eliminare gli oggetti memorizzati nella`LiveEvents/Football`folder e la cartella`LiveEvents/Baseball`cartella dopo che hanno più 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 nel`Football`cartella corrispondente al modello`index*.m3u8`dopo che hanno più di 15 secondi. MediaStoreelimina 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. Hai impostato una data di scadenza e dopo la data di scadenza di AWS ElementalMediaStoreelimina 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. Apertura dellaMediaStoreConsole alhttps://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 comando `put-lifecycle-policy`.

```
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. Apertura dellaMediaStoreConsole alhttps://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 comando `get-lifecycle-policy`.

  ```
  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. Apertura dellaMediaStoreConsole alhttps://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 comando `put-lifecycle-policy`.

```
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. Apertura dellaMediaStoreConsole alhttps://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 comando `delete-lifecycle-policy`.

```
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 relative al ciclo di vita degli oggetti: Scadenza in pochi secondi
- Policy di esempio relative al ciclo di vita degli oggetti: Scadenza entro alcuni giorni
- Policy di esempio relative al ciclo di vita degli oggetti: Passaggio alla classe di archiviazione con accesso non frequente
- Policy di esempio relative al ciclo di vita degli oggetti: Regole multiple
- Policy di esempio relative al ciclo di vita degli oggetti: Container vuoto

## Policy di esempio relative al ciclo di vita degli oggetti: Scadenza in pochi secondi

La policy seguente consente a MediaStore di eliminare gli oggetti che corrispondono a 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 alcuni giorni

La policy seguente consente a MediaStore di eliminare gli oggetti che corrispondono a 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: Passaggio alla classe di archiviazione con accesso non frequente

La policy seguente specifica cheMediaStoreSposta gli oggetti nella classe di archiviazione con accesso non frequente (IA) quando hanno 30 giorni di età precedente. 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 relative al ciclo di vita degli oggetti: Regole multiple

La seguente policy specifica che MediaStore deve effettuare 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 relative al ciclo di vita degli oggetti: Container vuoto

La seguente policy del ciclo di vita degli oggetti specifica che MediaStore deve eliminare tutti gli oggetti nel container, incluse cartelle e sottocartelle, 1 giorno dopo l'aggiunta al container. Se il container contiene oggetti prima dell'applicazione di questa policy,MediaStoreelimina gli oggetti 1 giorno dopo l'entrata in vigore della policy. 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"
        }
    ]
}
```

# Politiche metriche in AWS Elemental MediaStore

Per ogni contenitore, puoi aggiungere una policy metrica per consentire ad AWS MediaStore Elemental di inviare metriche ad Amazon CloudWatch. Affinché la nuova policy diventi effettiva, sono necessari fino a 20 minuti. Per una descrizione di ogni MediaStore metrica, consultaMediaStore 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 corrisponde a più regole, CloudWatch visualizza un punto dati per ogni regola corrispondente. Ad esempio, se un oggetto corrisponde a due regole denominate`rule1` e`rule2`, 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 politica metrica contiene regole che stabiliscono quali metriche AWS Elemental MediaStore invia ad Amazon CloudWatch. Per esempi di policy di parametro, consulta Policy di parametro di esempio.

Per aggiungere una policy di parametro (console)

1. Apri la MediaStore console all'indirizzo 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 per visualizzare la policy di parametro di un container.

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 politica metrica contiene regole che stabiliscono quali metriche AWS Elemental MediaStore invia ad Amazon 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 MediaStore console all'indirizzo 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

Questa politica di esempio indica che AWS Elemental MediaStore deve inviare metriche ad Amazon CloudWatch a 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 politica, 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

Questa politica di esempio indica che AWS Elemental non MediaStore deve inviare metriche ad Amazon CloudWatch a 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

Questa politica di esempio indica che AWS Elemental MediaStore deve inviare metriche ad Amazon CloudWatch a livello di container. Inoltre, MediaStore dovrebbe inviare le metriche 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`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

Questa politica di esempio indica che AWS Elemental MediaStore deve inviare metriche ad Amazon CloudWatch a livello di container. Inoltre, MediaStore dovrebbe inviare anche metriche 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

Questa politica di esempio indica che AWS Elemental MediaStore deve inviare metriche ad Amazon CloudWatch a livello di container. Inoltre, MediaStore dovrebbe inviare metriche per due cartelle:`sports/football/saturday` `esports/football`.

Le metriche relative MediaStore alle 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 ElementalMediaStore

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 ElementalMediaStorearchivia l'oggetto nella cartella. Se la cartella non esiste, il servizio la crea e quindi archivia l'oggetto nella cartella.

Supponiamo ad esempio che tu abbia un container denominato`moviese` carichi un file denominato`mlaw.ts`con il percorso`premium/canada`. AWS ElementalMediaStorearchivia l'oggetto nella sottocartella "canada" 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 ElementalMediaStoreUna volta eliminato l'ultimo oggetto in una cartella, elimina automaticamente la 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), minuscole (a-z), numeri (0-9), punti (.), trattini (-), tilde (~), caratteri di sottolineatura (_), segni di uguale (=) e due punti (:).
- Il nome deve essere composto da almeno un carattere. Nomi di cartelle vuote (come ad esempio`folder1//folder3/`) non sono ammessi.
- 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 ElementalMediaStorearchivia l'oggetto nella cartella. Se la cartella non esiste, il servizio la crea e quindi archivia l'oggetto nella cartella.

Per ulteriori informazioni, consultare . 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 ElementalMediaStoreUna volta eliminato l'ultimo oggetto in una cartella, elimina automaticamente la 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, consultare Eliminazione di un oggetto.

# Oggetti in AWS ElementalMediaStore

AWS ElementalMediaStorele risorse di sono chiamateoggetti. 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. Devi creare gli oggetti in locale prima di poterli caricare in MediaStore.
- Download (Scarica): copiare un oggetto da MediaStore in un'altra posizione. Questa operazione non elimina 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 ElementalMediaStorearchivia 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 ElementalMediaStore.

Puoi utilizzare la console di MediaStore o AWS CLI per caricare gli oggetti.

MediaStore supporta il trasferimento a blocchi di oggetti, che consente di ridurre 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 specifichi questa intestazione nella richiesta, MediaStore assegna il valore predefinito di `standard` per la disponibilità di caricamento 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 di file di oggetti possono contenere solo lettere, numeri, punti (.), trattini bassi (_), tilde (~), trattini (-), segni di uguale (=) e virgole (:).

Per caricare un oggetto (console)

1. Apertura dellaMediaStoreConsole inhttps://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 comando `put-object`. 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

È possibile utilizzare AWS ElementalMediaStoreConsole per visualizzare gli elementi (oggetti e cartelle) memorizzati nel livello principale di un container o in una cartella. Gli elementi archiviati in una sottocartella del container o della cartella corrente non verranno visualizzati. Puoi utilizzare AWS CLI per visualizzare un elenco di oggetti e cartelle all'interno di un container, indipendentemente dal numero di cartelle o sottocartelle presenti all'interno del container.

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

1. Apertura dellaMediaStoreConsole inhttps://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. Apertura dellaMediaStoreConsole inhttps://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 comando `list-items`.

```
aws mediastore-data list-items --endpoint https://
aaabbbcccdddee.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)

- In AWS CLI, utilizza il comando `list-items` 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 ElementalMediaStorearchivia i dettagli quali la data di modifica, la lunghezza del contenuto, l'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. Apertura dellaMediaStoreConsole inhttps://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 comando `describe-object`.

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

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 utilizzare AWS CLI per scaricare un oggetto intero o solo una parte.

Per scaricare un oggetto (console)

1. Apertura dellaMediaStoreConsole inhttps://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 comando `get-object`.

```
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, utilizza il comando `get-object` e specifica un intervallo.

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

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 ElementalMediaStoreoffre diverse opzioni per eliminare oggetti dai container:

- 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 ElementalMediaStoreelimina automaticamente la cartella e tutte cartelle vuote ai livelli superiori della 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. Apertura dellaMediaStoreConsole inhttps://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 comando `delete-object`.

  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. Apertura dellaMediaStoreConsole inhttps://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. Confermare che si desidera svuotare il container immettendo il nome del container nel campo di testo, quindi scegliereEmpty (Vuoto).

# Sicurezza in Elemental AWS MediaStore

Sicurezza nel cloud presso AWS è la massima priorità. Come un AWS cliente, trae vantaggio da data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS e tu. 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 in esecuzione AWS 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 del AWS Programmi di conformità . Per ulteriori informazioni sui programmi di conformità che si applicano a AWS Elemental MediaStore, consulta AWS Servizi rientranti nell'ambito del programma di conformità .
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

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

Argomenti

# Protezione dei dati in AWS Elemental MediaStore

Il AWS modello di responsabilità condivisa modello di di si applica alla protezione dei dati in AWS MediaStore Elemental. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutte le Cloud AWS. L'utente è responsabile del mantenimento del controllo sui contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile delle attività di configurazione e gestione della sicurezza per Servizi AWS che usi. Per ulteriori informazioni sulla privacy dei dati, consulta la sezione Privacy dei dati FAQ. Per informazioni sulla protezione dei dati in Europa, consulta la AWS Modello di responsabilità condivisa e post sul GDPR blog sul AWS Blog sulla sicurezza.

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

- Usa l'autenticazione a più fattori (MFA) con ogni account.
- UsaSSL/TLSper comunicare con AWS risorse. Richiediamo TLS 1.2 e consigliamo TLS 1.3.
- Configurazione API e registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi di acquisizione AWS attività, vedi Lavorare con i CloudTrail sentieri in AWS CloudTrail Guida per l'utente.
- Utilizzo AWS soluzioni di 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 sono necessari FIPS 140-3 moduli crittografici convalidati per l'accesso AWS tramite un'interfaccia a riga di comando o unAPI, utilizza un endpoint. FIPS Per ulteriori informazioni sugli FIPS endpoint disponibili, vedere Federal Information Processing Standard (FIPS) 140-3.

Ti consigliamo vivamente 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 con MediaStore o altro Servizi AWS utilizzando la consoleAPI, AWS CLI, oppure 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. Se fornisci un URL a un server esterno, ti consigliamo vivamente di non includere le informazioni sulle credenziali URL per convalidare la tua richiesta a quel server.

## Crittografia dei dati

MediaStore crittografa contenitori e oggetti inattivi utilizzando l'algoritmo -256 standard AES di 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 le origini (CORS) per consentire l'accesso selettivo tra origini diverse alle tue risorse. MediaStore ConCORS, puoi consentire alle applicazioni web client caricate in un dominio di interagire con le risorse di un dominio diverso. 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 Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso a AWS risorse. IAMgli amministratori controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (dispone delle autorizzazioni) a utilizzare le risorse. MediaStore IAMè un Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

- Destinatari
- Autenticazione con identità
- Gestione dell'accesso con policy
- Come MediaStore funziona AWS Elemental con IAM
- Esempi di politiche basate sull'identità per Elemental AWS MediaStore
- Risoluzione dei problemi relativi all'identità e all'accesso di AWS Elemental MediaStore

## Destinatari

Come si usa AWS Identity and Access Management (IAM) differisce a seconda del lavoro svolto. MediaStore

Utente del servizio: se utilizzi il MediaStore servizio per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più MediaStore

funzionalità per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità in MediaStore, consultaRisoluzione dei problemi relativi all'identità e all'accesso di AWS Elemental MediaStore .

Amministratore del servizio: se sei responsabile delle MediaStore risorse della tua azienda, probabilmente hai pieno accesso a MediaStore. È tuo compito determinare a quali MediaStore funzionalità e risorse devono accedere gli utenti del servizio. È quindi necessario inviare richieste all'IAMamministratore per modificare le autorizzazioni degli utenti del servizio. Consulta le informazioni contenute in questa pagina per comprendere i concetti di base diIAM. Per ulteriori informazioni su come la tua azienda può utilizzare IAM con MediaStore, consultaCome MediaStore funziona AWS Elemental con IAM.

IAMamministratore: se sei un IAM amministratore, potresti voler conoscere i dettagli su come scrivere politiche a cui gestire l'accesso MediaStore. Per visualizzare esempi di policy MediaStore basate sull'identità che puoi utilizzare inIAM, consulta. Esempi di politiche basate sull'identità per Elemental AWS MediaStore

## Autenticazione con identità

L'autenticazione è la modalità di accesso a AWS utilizzando le tue credenziali di identità. Devi essere autenticato (aver effettuato l'accesso a AWS) come Utente root dell'account AWS, come IAM utente o assumendo un IAM ruolo.

Puoi accedere a AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAMIdentity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali Google o Facebook sono esempi di identità federate. Quando accedi come identità federata, l'amministratore aveva precedentemente configurato la federazione delle identità utilizzando i ruoli. IAM Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente che sei, puoi accedere a AWS Management Console o il AWS portale di accesso. Per ulteriori informazioni sull'accesso a AWS, vedi Come accedere al tuo Account AWS nella Accedi ad AWS Guida per l'utente.

Se accedi AWS programmaticamente, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le richieste utilizzando le credenziali dell'utente. Se non usi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni

sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, vedi Firma AWS APIrichieste nella Guida IAM per l'utente.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del proprio account. Per ulteriori informazioni, consulta Autenticazione a più fattori nel AWS IAM Identity Center Guida per l'utente e utilizzo dell'autenticazione a più fattori () MFA in AWS nella Guida per l'utente di IAM.

### Account AWS utente root

Quando crei un Account AWS, inizi con un'unica identità di accesso con accesso completo a tutti Servizi AWS e le risorse presenti nell'account. Questa identità è denominata Account AWS utente root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per l'elenco completo delle attività che richiedono l'accesso come utente root, consulta Attività che richiedono le credenziali dell'utente root nella Guida per l'IAMutente.

### Identità federata

Come procedura ottimale, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, un provider di identità Web, il AWS Directory Service, la directory Identity Center o qualsiasi utente che accede Servizi AWS utilizzando le credenziali fornite tramite una fonte di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, si consiglia di utilizzare AWS IAM Identity Center. È possibile creare utenti e gruppi in IAM Identity Center oppure connettersi e sincronizzarsi con un set di utenti e gruppi nella propria fonte di identità per utilizzarli su tutti i Account AWS e applicazioni. Per informazioni su IAM Identity Center, vedi Cos'è IAM Identity Center? nel AWS IAM Identity Center Guida per l'utente.

### IAM users and groups

Un IAMutente è un'identità all'interno del tuo Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Laddove possibile, consigliamo di fare affidamento su

credenziali temporanee anziché creare IAM utenti con credenziali a lungo termine come password e chiavi di accesso. Tuttavia, se hai casi d'uso specifici che richiedono credenziali a lungo termine con IAM gli utenti, ti consigliamo di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta Ruotare regolarmente le chiavi di accesso per i casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente. IAM

Un IAMgruppo è un'identità che specifica un insieme di utenti. IAM Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, potresti avere un gruppo denominato IAMAdminse concedere a quel gruppo le autorizzazioni per IAM amministrare le risorse.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta Quando creare un IAM utente (anziché un ruolo) nella Guida per l'IAMutente.

### IAMruoli

Un IAMruolo è un'identità all'interno del tuo Account AWS che dispone di autorizzazioni specifiche. È simile a un IAM utente, ma non è associato a una persona specifica. È possibile assumere temporaneamente un IAM ruolo nel AWS Management Console cambiando ruolo. Puoi assumere un ruolo chiamando un AWS CLI oppure AWS APIoperazione o utilizzando un comando personalizzatoURL. Per ulteriori informazioni sui metodi di utilizzo dei ruoli, vedere Utilizzo IAM dei ruoli nella Guida per l'IAMutente.

IAMi ruoli con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per informazioni sui ruoli per la federazione, vedere Creazione di un ruolo per un provider di identità di terze parti nella Guida per l'IAMutente. Se utilizzi IAM Identity Center, configuri un set di autorizzazioni. Per controllare a cosa possono accedere le identità dopo l'autenticazione, IAM Identity Center correla il set di autorizzazioni a un ruolo in. IAM Per informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella AWS IAM Identity Center Guida per l'utente.
- Autorizzazioni IAM utente temporanee: un IAM utente o un ruolo può assumere un IAM ruolo per assumere temporaneamente autorizzazioni diverse per un'attività specifica.

- Accesso su più account: puoi utilizzare un IAM ruolo per consentire a qualcuno (un responsabile fidato) di un altro account di accedere alle risorse del tuo account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per conoscere la differenza tra i ruoli e le politiche basate sulle risorse per l'accesso tra più account, consulta l'accesso alle risorse tra account IAM nella Guida per l'utente. IAM
- Accesso a più servizi: alcuni Servizi AWS usa le funzionalità in altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è normale che quel servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
  - Sessioni di accesso diretto (FAS): quando utilizzi un IAM utente o un ruolo per eseguire azioni in AWS, sei considerato un preside. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FASutilizza le autorizzazioni del principale che chiama un Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi a valle. FASle richieste vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse da completare. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli FAS delle politiche relative alle richieste, consulta Forward access sessions.
  - Ruolo di servizio: un ruolo di servizio è un IAMruolo che un servizio assume per eseguire azioni per conto dell'utente. Un IAM amministratore può creare, modificare ed eliminare un ruolo di servizio dall'internoIAM. Per ulteriori informazioni, vedere Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS nella Guida per l'utente di IAM.
  - Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo di eseguire un'azione per conto dell'utente. I ruoli collegati ai servizi vengono visualizzati nel Account AWS e sono di proprietà del servizio. Un IAM amministratore può visualizzare, ma non modificare le autorizzazioni per i ruoli collegati al servizio.
- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un IAM ruolo per gestire le credenziali temporanee per le applicazioni in esecuzione su un'EC2istanza e in fase di creazione AWS CLI oppure AWS APIrichieste. Ciò è preferibile alla memorizzazione delle chiavi di accesso all'interno dell'EC2istanza. Per assegnare un AWS assegnare un ruolo a un'EC2istanza e renderlo disponibile a tutte le relative applicazioni, è necessario creare un profilo di istanza collegato all'istanza. Un profilo di istanza contiene il ruolo e consente ai programmi in esecuzione sull'EC2istanza di ottenere credenziali temporanee. Per ulteriori informazioni, consulta Usare un IAM ruolo per

concedere le autorizzazioni alle applicazioni in esecuzione su EC2 istanze Amazon nella Guida per l'IAMutente.

Per sapere se utilizzare IAM ruoli o IAM utenti, consulta Quando creare un IAM ruolo (anziché un utente) nella Guida per l'IAMutente.

## Gestione dell'accesso con policy

Puoi controllare l'accesso in AWS creando politiche e allegandole a AWS identità o risorse. Una politica è un oggetto in AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata in AWS come JSON documenti. Per ulteriori informazioni sulla struttura e il contenuto dei documenti relativi alle JSON politiche, vedere Panoramica delle JSON politiche nella Guida per l'IAMutente.

Gli amministratori possono utilizzare AWS JSONpolitiche per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti il permesso di eseguire azioni sulle risorse di cui hanno bisogno, un IAM amministratore può creare IAM politiche. L'amministratore può quindi aggiungere le IAM politiche ai ruoli e gli utenti possono assumerli.

IAMle politiche definiscono le autorizzazioni per un'azione indipendentemente dal metodo utilizzato per eseguire l'operazione. Ad esempio, supponiamo di disporre di una policy che consente l'operazione `iam:GetRole`. Un utente con tale criterio può ottenere informazioni sul ruolo da AWS Management Console, il AWS CLI, o il AWS API.

### Policy basate su identità

I criteri basati sull'identità sono documenti relativi ai criteri di JSON autorizzazione che è possibile allegare a un'identità, ad esempio un IAM utente, un gruppo di utenti o un ruolo. Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una politica basata sull'identità, consulta Creazione di politiche nella Guida per l'utente. IAM IAM

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche

gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono AWS politiche gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una politica gestita o una politica in linea, consulta Scelta tra politiche gestite e politiche in linea nella Guida per l'IAMutente.

### Policy basate su risorse

Le politiche basate sulle risorse sono documenti di JSON policy allegati a una risorsa. Esempi di politiche basate sulle risorse sono le policy di trust dei IAM ruoli 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 azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o Servizi AWS.

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi usare AWS politiche gestite da IAM una politica basata sulle risorse.

### Liste di controllo degli accessi () ACLs

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

Amazon S3, AWS WAF e Amazon VPC sono esempi di servizi che supportanoACLs. Per ulteriori informazioniACLs, consulta la panoramica di Access control list (ACL) nella Amazon Simple Storage Service Developer Guide.

### Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite di autorizzazioni è una funzionalità avanzata in cui si impostano le autorizzazioni massime che una politica basata sull'identità può concedere a un'entità (utente o ruolo). IAM IAM È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal`sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di

queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità nella Guida per l'utente. IAM IAM

- Politiche di controllo del servizio (SCPs): SCPs sono JSON politiche che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in AWS Organizations. AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più Account AWS di proprietà della tua azienda. Se abiliti tutte le funzionalità di un'organizzazione, puoi applicare le politiche di controllo del servizio (SCPs) a uno o tutti i tuoi account. I SCP limiti e le autorizzazioni per le entità presenti negli account dei membri, inclusi tutti Utente root dell'account AWS. Per ulteriori informazioni su Organizations andSCPs, vedere Service control policies nel AWS Organizations Guida per l'utente.
- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta le politiche di sessione nella Guida IAM per l'utente.

### Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come AWS determina se consentire una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle policy nella Guida per l'IAMutente.

## Come MediaStore funziona AWS Elemental con IAM

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

IAMfunzionalità che puoi usare con AWS Elemental MediaStore

| IAMcaratteristica | MediaStore supporto |
|---|---|
| Policy basate su identità | Sì |
| Policy basate su risorse | Sì |

| IAMcaratteristica | MediaStore supporto |
| --- | --- |
| Azioni di policy | Sì |
| Risorse relative alle policy | Sì |
| Chiavi di condizione della policy (specifica del servizio) | Sì |
| ACLs | No |
| ABAC(tag nelle politiche) | Parziale |
| Credenziali temporanee | Sì |
| Autorizzazioni del principale | Sì |
| Ruoli di servizio | Sì |
| Ruoli collegati al servizio | No |

Per avere una visione di alto livello di come MediaStore e altro AWS i servizi funzionano con la maggior parte delle IAM funzionalità, vedi AWS servizi compatibili con IAM la Guida per l'IAMutente.

## Politiche basate sull'identità per MediaStore

Supporta le policy basate su identità: sì

Le politiche basate sull'identità sono documenti relativi alle politiche di JSON autorizzazione che è possibile allegare a un'identità, ad esempio un IAM utente, un gruppo di utenti o un ruolo. Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una politica basata sull'identità, consulta Creazione di politiche nella Guida per l'utente. IAM IAM

Con le politiche IAM basate sull'identità, puoi specificare azioni e risorse consentite o negate, nonché le condizioni in base alle quali le azioni sono consentite o negate. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per ulteriori informazioni su tutti gli elementi che è possibile utilizzare in una JSON politica, vedere il riferimento agli elementi IAM JSON della politica nella Guida per l'IAMutente.

Esempi di policy basate sull'identità per MediaStore

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

## Politiche basate sulle risorse all'interno MediaStore

Supporta politiche basate sulle risorse: Sì

Le politiche basate sulle risorse sono documenti di JSON policy allegati a una risorsa. Esempi di politiche basate sulle risorse sono le policy di trust dei IAM ruoli 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 azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o Servizi AWS.

Per abilitare l'accesso tra più account, puoi specificare un intero account o IAM entità in un altro account come principale in una politica basata sulle risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un IAM amministratore dell'account affidabile deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta Cross Account Resource Access IAM nella Guida IAM per l'utente.

> **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 AWS JSONpolitiche per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'`Action`elemento di una JSON policy descrive le azioni che è possibile utilizzare per consentire o negare l'accesso a una policy. Le azioni politiche in genere hanno lo stesso nome di quelle associate AWS APIoperazione. Esistono alcune eccezioni, come le azioni di sola autorizzazione che non hanno un'operazione corrispondente. API Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi 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 politiche basate sull'identità per Elemental AWS MediaStore

## Risorse politiche per MediaStore

Supporta le risorse di policy: sì

Gli amministratori possono utilizzare AWS JSONpolitiche per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Resource` JSON policy specifica l'oggetto o gli oggetti a cui si applica l'azione. Le istruzioni devono includere un elemento `Resource`o un elemento `NotResource`. Come best

practice, specifica una risorsa utilizzando il relativo Amazon Resource Name (ARN). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

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

Per visualizzare un elenco di tipi di MediaStore risorse e relativiARNs, consulta Resources defined by AWS Elemental MediaStore nel Service Authorization Reference. Per sapere con quali azioni puoi specificare le caratteristiche ARN di ogni risorsa, vedi Azioni definite da AWS Elemental MediaStore.

La risorsa MediaStore contenitore ha quanto segue: ARN

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

Per ulteriori informazioni sul formato diARNs, consulta Amazon Resource Names (ARNs) e AWS Namespace dei servizi.

Ad esempio, per specificare il `AwardsShow` contenitore nella dichiarazione, utilizza quanto segue: ARN

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

## Chiavi relative alle condizioni della politica per MediaStore

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

Gli amministratori possono utilizzare AWS JSONpolitiche per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Condition`(o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition`è facoltativo. Puoi 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.

Se specificate più `Condition` elementi in un'istruzione o più chiavi in un singolo `Condition` elemento, AWS li valuta utilizzando un'`AND`operazione logica. Se specificate più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'`OR`operazione logica. Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, è possibile concedere a un IAM utente l'autorizzazione ad accedere a una risorsa solo se è contrassegnata con il relativo nome IAM utente. Per ulteriori informazioni, consulta gli elementi IAM della politica: variabili e tag nella Guida IAM per l'utente.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per vedere tutto AWS chiavi di condizione globali, vedi AWS chiavi di contesto della condizione globale nella Guida IAM per l'utente.

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 usare una chiave di condizione, vedi Azioni definite da AWS Elemental MediaStore.

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

## ACLsin MediaStore

SupportiACLs: 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. ACLssono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy. JSON

## ABACcon MediaStore

Supporti ABAC (tag nelle politiche): Parziale

Il controllo degli accessi basato sugli attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. È possibile allegare tag a IAM entità (utenti o ruoli) e a molte altre AWS risorse. L'etichettatura di entità e risorse è il primo passo diABAC. Quindi si progettano ABAC politiche per consentire le operazioni quando il tag del principale corrisponde al tag sulla risorsa a cui sta tentando di accedere.

ABACè utile in ambienti in rapida crescita e aiuta in situazioni in cui la gestione delle politiche diventa complicata.

Per controllare l'accesso basato su tag, fornisci 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 Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni suABAC, vedere Cos'è? ABAC nella Guida IAM per l'utente. Per visualizzare un tutorial con i passaggi per la configurazioneABAC, consulta Utilizzare il controllo di accesso basato sugli attributi (ABAC) nella Guida per l'IAMutente.

## Utilizzo di credenziali temporanee con MediaStore

Supporta le credenziali temporanee: sì

Medio Servizi AWS non funzionano quando accedi utilizzando credenziali temporanee. Per ulteriori informazioni, tra cui Servizi AWS lavorare con credenziali temporanee, vedere Servizi AWS che funzionano con IAM la Guida per l'IAMutente.

Stai utilizzando credenziali temporanee se accedi a AWS Management Console utilizzando qualsiasi metodo tranne il nome utente e la password. Ad esempio, quando accedi AWS utilizzando il link Single Sign-On (SSO) della vostra azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sul cambio di ruolo, consulta Passare a un ruolo (console) nella Guida per l'IAMutente.

È possibile creare manualmente credenziali temporanee utilizzando AWS CLI oppure AWS API. È quindi possibile utilizzare tali credenziali temporanee per accedere AWS. AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, vedere Credenziali di sicurezza temporanee in. IAM

## Autorizzazioni principali multiservizio per MediaStore

Supporta sessioni di accesso diretto (FAS): Sì

Quando si utilizza un IAM utente o un ruolo per eseguire azioni in AWS, sei considerato un preside. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FASutilizza le autorizzazioni del principale che chiama un Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi a valle. FASle richieste

vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse da completare. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli FAS delle politiche relative alle richieste, consulta Forward access sessions.

### Ruoli di servizio per MediaStore

Supporta i ruoli di servizio: sì

Un ruolo di servizio è un IAMruolo che un servizio assume per eseguire azioni per conto dell'utente. Un IAM amministratore può creare, modificare ed eliminare un ruolo di servizio dall'internoIAM. Per ulteriori informazioni, vedere Creazione di un ruolo per delegare le autorizzazioni a un 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 di eseguire un'azione per conto dell'utente. I ruoli collegati ai servizi vengono visualizzati nel Account AWS e sono di proprietà del servizio. Un IAM amministratore può visualizzare, ma non modificare le autorizzazioni per i ruoli collegati al servizio.

Per informazioni dettagliate sulla creazione o la gestione di ruoli collegati ai servizi, vedere AWS servizi che funzionano con. 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 politiche basate sull'identità per Elemental AWS MediaStore

Per impostazione predefinita, gli utenti e i ruoli non sono autorizzati a creare o modificare risorse. MediaStore Inoltre, non possono eseguire attività utilizzando il AWS Management Console, AWS

Command Line Interface (AWS CLI) o AWS API. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un IAM amministratore può creare IAM politiche. L'amministratore può quindi aggiungere le IAM politiche ai ruoli e gli utenti possono assumerli.

Per informazioni su come creare una politica IAM basata sull'identità utilizzando questi documenti di esempioJSON, consulta Creazione di IAM politiche nella Guida per l'IAMutente.

Per i 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 MediaStore console
- 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 azioni possono comportare costi per Account AWS. Quando crei o modifichi politiche basate sull'identità, segui queste linee guida e consigli:

- Inizia con AWS politiche gestite e passaggio alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza il AWS politiche gestite che concedono autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo Account AWS. Si consiglia di ridurre ulteriormente le autorizzazioni definendo AWS politiche gestite dai clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta AWS politiche gestite o AWS politiche gestite per le funzioni lavorative nella Guida per IAM l'utente.
- Applica le autorizzazioni con privilegi minimi: quando imposti le autorizzazioni con le IAM politiche, concedi solo le autorizzazioni necessarie per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo per applicare le autorizzazioni, consulta Politiche e autorizzazioni nella Guida IAM per l'utente. IAM IAM
- Utilizza le condizioni nelle IAM politiche per limitare ulteriormente l'accesso: puoi aggiungere una condizione alle tue politiche per limitare l'accesso ad azioni e risorse. Ad esempio, puoi scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzandoSSL. È inoltre possibile utilizzare le condizioni per concedere l'accesso alle azioni di servizio se vengono

utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta Elementi IAM JSON della politica: Condizione nella Guida IAM per l'utente.

- Usa IAM Access Analyzer per convalidare IAM le tue policy e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano al linguaggio delle IAM policy () e alle best practice. JSON IAM IAMAccess Analyzer fornisce più di 100 controlli delle politiche e consigli pratici per aiutarti a creare policy sicure e funzionali. Per ulteriori informazioni, vedere Convalida delle policy di IAM Access Analyzer nella Guida per l'utente. IAM
- Richiedi l'autenticazione a più fattori (MFA): se si dispone di uno scenario che richiede IAM utenti o un utente root nel Account AWS, attivala MFA per una maggiore sicurezza. Per richiedere MFA quando vengono richiamate API le operazioni, aggiungi MFA delle condizioni alle tue politiche. Per ulteriori informazioni, vedere Configurazione dell'APIaccesso MFA protetto nella Guida per l'IAMutente.

Per ulteriori informazioni sulle procedure consigliate inIAM, consulta la sezione Procedure consigliate in materia di sicurezza IAM nella Guida per l'IAMutente.

## Utilizzo della MediaStore console

Per accedere a AWS MediaStore Console Elemental, devi disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse del MediaStore tuo Account AWS. Se crei una politica basata sull'identità che è più restrittiva delle autorizzazioni minime richieste, la console non funzionerà come previsto per le entità (utenti o ruoli) con quella politica.

Non è necessario consentire autorizzazioni minime per la console per gli utenti che effettuano chiamate solo verso AWS CLI o AWS API. Consenti invece l'accesso solo alle azioni che corrispondono all'APIoperazione che stanno cercando di eseguire.

Per garantire che gli utenti e i ruoli possano continuare a utilizzare la MediaStore console, collega anche MediaStore `ConsoleAccess` o `ReadOnly` AWS politica gestita alle entità. Per ulteriori informazioni, consulta Aggiungere autorizzazioni a un utente nella Guida per l'IAMutente.

## Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra come è possibile creare una politica che consenta IAM agli utenti di visualizzare le politiche in linea e gestite allegate alla loro identità utente. Questa politica include le

autorizzazioni per completare questa azione sulla console o utilizzando programmaticamente il AWS CLI oppure AWS API.

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

# Risoluzione dei problemi relativi all'identità e all'accesso di AWS Elemental MediaStore

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

Argomenti



## Non sono autorizzato a eseguire alcuna azione in MediaStore

Se ricevi un errore che indica che non disponi dell'autorizzazione per eseguire un'operazione, le tue policy devono essere aggiornate in modo che ti sei consentito eseguire tale operazione.

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

```
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 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 messaggio di errore indicante che non sei autorizzato a eseguire l'`iam:PassRole`azione, le tue politiche devono essere aggiornate per consentirti di assegnare un ruolo a MediaStore.

Medio Servizi AWS consentono di trasferire un ruolo esistente a quel servizio anziché 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 IAM utente denominato `marymajor` tenta di utilizzare la console per eseguire un'azione in. MediaStore Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare 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 AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

## Voglio consentire l'accesso a persone esterne al mio Account AWS per accedere 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 agli utenti l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se MediaStore supporta queste funzionalità, consulta. Come MediaStore funziona AWS Elemental con IAM
- Per scoprire come fornire l'accesso alle tue risorse in tutto il mondo Account AWS di cui sei proprietario, vedi Fornire l'accesso a un IAM utente in un altro Account AWS che possiedi nella Guida per l'IAMutente.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, vedi Fornire l'accesso a Account AWS di proprietà di terzi nella Guida per l'IAMutente.
- 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'IAMutente.
- Per conoscere la differenza tra l'utilizzo di ruoli e politiche basate sulle risorse per l'accesso tra account diversi, consulta la sezione Accesso alle risorse tra account nella Guida per l'utente. IAM IAM

# Registrazione e monitoraggio AWS Elemental MediaStore

Questa sezione fornisce una panoramica delle opzioni per la registrazione e il monitoraggio AWS Elemental MediaStore per motivi di sicurezza. Per ulteriori informazioni sulla registrazione e il monitoraggio MediaStore, vedereMonitoraggio e etichettatura in AWS Elemental MediaStore.

Il monitoraggio è una parte importante del mantenimento dell'affidabilità, della disponibilità e delle prestazioni di AWS Elemental MediaStore e il tuo AWS soluzioni. Dovresti raccogliere dati di monitoraggio da tutte le parti del tuo AWS soluzione che consente di eseguire più facilmente il debug di un errore multipunto, se ne verifica uno. 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 SNS argomento di Amazon o a una politica di Auto AWS 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 log

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 ulteriori dettagli. Per ulteriori informazioni, consulta Registrazione delle chiamate API con CloudTrail.

## AWS Trusted Advisor

Trusted Advisor attinge alle migliori pratiche apprese servendo centinaia di migliaia di AWS clienti. Trusted Advisor ispeziona l'AWSambiente e quindi 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 AWS i clienti hanno accesso a cinque controlli Trusted Advisor. I clienti con un piano di supporto Business o Enterprise possono visualizzarli tutti Trusted Advisor assegni.

Per ulteriori informazioni, consulta AWS Trusted Advisor.

# Convalida della conformità per Elemental AWS MediaStore

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

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

La tua 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. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Guide introduttive su sicurezza e conformità: queste guide all'implementazione illustrano le considerazioni relative all'architettura e forniscono i passaggi per l'implementazione degli ambienti di base su AWS incentrati sulla sicurezza e la conformità.
- Architettura per la HIPAA sicurezza e la conformità su Amazon Web Services: questo white paper descrive come le aziende possono utilizzare AWS per creare applicazioni idonee. HIPAA

  > **Note**
  >
  > Non tutti Servizi AWS sono HIPAA idonei. Per ulteriori informazioni, consulta la Guida ai servizi HIPAA idonei.

- AWS Risorse per la conformità : questa raccolta di cartelle di lavoro e guide potrebbe riguardare il tuo settore e la tua località.
- AWS Guide alla conformità dei clienti: comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione Servizi AWS e mappano le linee guida per i controlli di sicurezza su più framework (tra cui il National Institute of Standards and Technology (NIST), il Payment Card Industry Security Standards Council (PCI) e l'International Organization for Standardization ()). ISO
- Valutazione delle risorse con regole in AWS Config Guida per gli sviluppatori: la AWS Config il servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- AWS Security Hub— Questo Servizio AWS fornisce una visione completa dello stato di sicurezza all'interno AWS. Security Hub utilizza i controlli di sicurezza per valutare i tuoi AWS risorse e per verificare la vostra conformità agli standard e alle migliori pratiche del settore della sicurezza. Per

un elenco dei servizi e dei controlli supportati, consulta la pagina Documentazione di riferimento sui controlli della Centrale di sicurezza.

- Amazon GuardDuty — Questo Servizio AWS rileva potenziali minacce per il tuo Account AWS, carichi di lavoro, contenitori e dati monitorando l'ambiente alla ricerca di attività sospette e dannose. GuardDuty può aiutarti a soddisfare vari requisiti di conformità, ad esempio PCI DSS soddisfacendo i requisiti di rilevamento delle intrusioni imposti da determinati framework di conformità.
- AWS Audit Manager— Questo Servizio AWS ti aiuta a controllare continuamente i tuoi AWS utilizzo per semplificare la gestione del rischio e la conformità alle normative e agli standard di settore.

## Resilienza in Elemental AWS MediaStore

Il AWS l'infrastruttura globale è costruita attorno Regioni AWS e 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, puoi 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 sull' Regioni AWS e zone di disponibilità, vedi AWS Infrastruttura globale.

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

## Sicurezza dell'infrastruttura in Elemental AWS MediaStore

In quanto servizio gestito, AWS Elemental MediaStore è protetto da AWS sicurezza di rete globale. Per informazioni su AWS servizi di sicurezza e come AWS protegge l'infrastruttura, vedi AWS Sicurezza nel cloud. Per progettare il tuo AWS ambiente che utilizza le migliori pratiche per la sicurezza dell'infrastruttura, vedi Infrastructure Protection in  Security Pillar AWS Framework ben architettato.

Tu usi AWS APIchiamate pubblicate per l'accesso MediaStore tramite la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). Richiediamo TLS 1.2 e consigliamo TLS 1.3.

- Suite di cifratura con Perfect Forward Secrecy (PFS) come (Ephemeral Diffie-Hellman) o DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale. IAM Oppure puoi usare il AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per firmare le richieste.

## 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. In AWS, l'impersonificazione tra diversi servizi può portare alla confusione del 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 che ciò accada, AWS fornisce strumenti che consentono di proteggere i dati per tutti i servizi con responsabili del servizio a cui è stato concesso l'accesso alle risorse del tuo account.

Ti consigliamo di utilizzare `aws:SourceArn`le chiavi di contesto della condizione `aws:SourceAccount`globale nelle politiche delle risorse per limitare le autorizzazioni che AWS Elemental MediaStore concede a un altro servizio alla risorsa. Utilizza `aws:SourceArn` se desideri consentire l'associazione di una sola risorsa all'accesso tra servizi. Utilizza `aws:SourceAccount` se desideri consentire l'associazione di qualsiasi risorsa in tale account all'uso tra servizi.

Il modo più efficace per proteggersi dal confuso problema del vice è utilizzare la chiave di contesto ARN della condizione `aws:SourceArn` globale con l'intera risorsa. Se non conosci la dimensione completa ARN della risorsa o se stai specificando più risorse, usa la chiave `aws:SourceArn` global context condition con caratteri jolly (`*`) per le parti sconosciute di. ARN Ad esempio `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Se il `aws:SourceArn` valore non contiene l'ID dell'account, ad esempio un bucket Amazon S3ARN, devi utilizzare entrambe le chiavi di contesto della condizione globale per limitare le autorizzazioni.

Il valore di `aws:SourceArn` deve essere la configurazione per la quale vengono MediaStore pubblicati CloudWatch i log nella regione e nell'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.

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

# Monitoraggio e etichettatura in AWS Elemental MediaStore

Il monitoraggio è importante per garantire l'affidabilità, la disponibilità e le prestazioni di AWS Elemental MediaStore e delle tue altreAWS soluzioni. AWSfornisce i seguenti strumenti di monitoraggio per controllare MediaStore, segnalare un problema e intervenire automaticamente quando necessario:

- AWS CloudTrail acquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo account AWS e fornisce i file di log a un bucket Amazon S3 specificato. Puoi identificare quali utenti e account hanno richiamato AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute. Per ulteriori informazioni, consultare la Guida per l'utente AWS CloudTrail.
- Amazon CloudWatch monitora leAWS risorse e le applicazioni che esegui suAWS in tempo reale. Puoi raccogliere i parametri e tenerne traccia, 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 impostare perché CloudWatch tenga traccia dell'uso della CPU o di altri parametri delle tue istanze Amazon EC2 e avviare automaticamente nuove istanze quando necessario. Per ulteriori informazioni, consulta la Guida per CloudWatch l'utente di Amazon.
- Amazon CloudWatch Events fornisce un flusso di eventi di sistema che descrivono le modifiche nelleAWS risorse. In genere,AWS i servizi forniscono notifiche di CloudWatch eventi a Events in pochi secondi, ma a volte può essere necessario un minuto o più. CloudWatch Events consente il calcolo automatizzato basato sugli eventi, così che tu possa scrivere le regole che osservano determinati eventi e attivano le operazioni automatizzate in altriAWS servizi quando si verificano gli eventi. Per ulteriori informazioni, consulta la Guida per l'utente di Amazon CloudWatch Events.
- Amazon CloudWatch Logs consente di monitorare, archiviare e accedere ai file di log dalle istanze Amazon EC2 e da altre origini. CloudTrail CloudWatch I log possono monitorare le informazioni nei file di log e notificare quando vengono raggiunte determinate soglie. Puoi inoltre archiviare i dati del log in storage estremamente durevole. Per ulteriori informazioni, consulta la Guida per l'utente CloudWatch di Amazon Logs.

Puoi anche assegnare metadati ai 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 usare i tag per organizzare le risorse AWS nella console di gestione AWS,

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

Argomenti



# Registrazione delle chiamate MediaStore API AWS Elemental conAWS CloudTrail

AWS Elemental MediaStore è integrato conAWS CloudTrail, un servizio che offre un record delle operazioni eseguite da un utente, un ruolo o unAWS servizio in MediaStore. CloudTrail acquisisce un sottoinsieme di chiamate API per MediaStore come eventi, incluse le chiamate dalla MediaStore console e dalle chiamate in codice all' MediaStore API. Se si crea un percorso, è possibile abilitare la distribuzione continua di CloudTrail eventi in un bucket Amazon S3, inclusi gli eventi per MediaStore. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console di in Cronologia eventi. Le informazioni raccolte da consentono CloudTrail di determinare la richiesta effettuata a MediaStore, l'indirizzo IP da cui è stata eseguita la richiesta, l'autore della richiesta, il momento in cui è stata eseguita e altro.

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

Argomenti



## MediaStoreInformazioni AWS Elemental in CloudTrail

CloudTrail è abilitato sull'AWSaccount al momento della sua creazione. Quando si verifica un'attività in AWS Elemental MediaStore, questa viene registrata in un CloudTrail evento insieme ad altri eventi delAWS servizio in Event history (Cronologia eventi). È possibile visualizzare, cercare e scaricare gli eventi recenti nell'account AWS. Per ulteriori informazioni, consulta Visualizzazione di eventi mediante la cronologia CloudTrail eventi di.

Per una registrazione continua degli eventi nell'account AWS che includa gli eventi per MediaStore, creare un trail. Un percorso consente di CloudTrail distribuire i file di log in 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 nella partizione AWS e distribuisce i file di log nel bucket Amazon S3 specificato. Inoltre, puoi configurare altriAWS servizi per analizzare con maggiore dettaglio e usare i dati raccolti nei CloudTrail log. Per ulteriori informazioni, consulta i seguenti argomenti:

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

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

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

Ogni evento o voce del 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 le 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 servizio AWS.

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

## Esempio: voci del file di MediaStore registro di AWS Elemental

Un trail è una configurazione che consente la distribuzione di eventi come i file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di log possono contenere una o più voci di log. Un evento rappresenta una singola richiesta da un'fonte e include informazioni sul operazione richiesta, data e ora dell'operazione, parametri richiesti e così via. CloudTrail i file di log non sono una traccia stack ordinata delle chiamate pubbliche dell'API, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail log di che illustra 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 MediaStore Elemental 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 un periodo di 15 mesi, per permettere l'accesso alle informazioni storiche e ottenere una prospettiva migliore sulle prestazioni del servizio o dell'applicazione 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 Guida per CloudWatch l'utente di Amazon.

AWSfornisce i seguenti strumenti di monitoraggio per controllare MediaStore, segnalare un problema e intervenire automaticamente quando necessario:

- Amazon CloudWatch Logs ti consente di monitorare, archiviare e accedere ai file di log daAWS servizi come AWS Elemental MediaStore. È possibile utilizzare CloudWatch i log per monitorare applicazioni e sistemi utilizzando i dati di log. Ad esempio, CloudWatch Logs è in grado di monitorare il numero di errori che si verificano nei registri delle applicazioni e di inviare una notifica ogni volta che il tasso di errori supera una soglia specificata. CloudWatch Logs utilizza i dati di log 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 il termine che cerchi viene trovato, CloudWatch Logs segnala i dati a una CloudWatch

metrica da te specificata. I dati di log vengono crittografati durante il transito e mentre sono a riposo.

- Amazon CloudWatch Events fornisce eventi di sistema che descrivono i cambiamenti nelleAWS risorse, come MediaStore gli oggetti. In genere,AWS i servizi forniscono notifiche di CloudWatch eventi a Events in pochi secondi, ma a volte può essere necessario un minuto o più. È possibile impostare regole per abbinare eventi (come una`DeleteObject` richiesta) e instradarli a una o più funzioni o flussi di destinazione. CloudWatch Gli eventi si accorgono delle modifiche operative appena si verificano. Inoltre, CloudWatch Events risponde a queste modifiche operative e prende le misure correttive necessarie inviando messaggi per rispondere all'ambiente attivando funzioni, effettuando modifiche e catturando 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 inoltre fornire informazioni sulla base clienti e sulla MediaStore fattura. CloudWatch I registri 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.

### Impostazione delle autorizzazioni per Amazon CloudWatch

UsaAWS Identity and Access Management (IAM) per creare un ruolo che dia ad AWS Elemental MediaStore l'accesso 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 MediaStore l'accesso 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:

```
{
    "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'internoAWS del tuo account.
4. Scegli Review policy (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. In Account ID (ID account) immettere l'ID dell'account AWS.
9. Scegliere 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 Edit trust relationship (Modifica relazione di trust).
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:

    ```
    {
      "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 default, AWS Elemental MediaStore non raccoglie i log di accesso. Quando si abilita la registrazione degli accessi in un container, MediaStore fornisce ad Amazon i log di accesso per gli oggetti memorizzati in quel container 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)

- In AWS CLI, usa il comando `start-access-logging`.

```
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 Elemental MediaStore 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)

- In 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

Quando i log di MediaStore accesso di AWS Elemental non vengono visualizzati in Amazon CloudWatch, consulta la tabella seguente per cause e risoluzioni potenziali.

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

| Sintomo | Il problema potrebbe essere... | Prova questa soluzione... |
| --- | --- | --- |
| Non viene visualizzato alcun CloudTrail evento, anche se CloudTrail i registri 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. Consultare the section called “Impostazione delle autorizzazioni per CloudWatch”. |
| Hai inviato una richiesta API `DescribeContainer` , ma la risposta mostra che il parametro `AccessLoggingEnabled` ha un valore di `False`. Inoltre, non visualizzi eventi CloudTrail per il ruolo `MediaStoreAccessLogs` quando effettui una chiamata `DescribeLogGroup` , `CreateLogGroup` , `DescribeLogStream` 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. Consultare the section called “Impostazione delle autorizzazioni per CloudWatch”. |
| | La registrazione degli accessi non è abilitata sul container. | Abilita i log di accesso per il container. Consultare the section called “Abilitazione della registrazione degli accessi”. |

| Sintomo | Il problema potrebbe essere... | Prova questa soluzione... |
| --- | --- | --- |
| 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:<br><br>`"eventSource": "logs.amazonaws.com",`<br><br>`"errorCode": "AccessDenied",`<br><br>`"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 AWS Elemental MediaStore. | Aggiorna il ruolo IAM per avere le autorizzazioni e la policy di attendibilità corretti. Consultare the section called “Impostazione delle autorizzazioni per CloudWatch”. |

| Sintomo | 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 registro per account per regione. Consulta le quote per i gruppi di log nella Amazon CloudWatch Logs User Guide. | Sulla CloudWatch console, stabilisci se il tuo account ha raggiunto la CloudWatc h quota per i gruppi di log. Se necessario, richiedere un aumento delle quote. |
| Vengono visualizzati alcuni log in CloudWatch, ma non tutti i log che ci si aspetta di vedere. | Il tuo account potrebbe aver superato la CloudWatch quota di transazioni al secondo per account per regione. Consulta le quote di cui`PutLogEve nts` alla guida per l'utente di Amazon CloudWatch Logs. | Richiedi un aumento della quota di CloudWatch transazio ni al secondo per account per 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'ID account AWS dell'account che è stato 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 unatransient 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 ha una regola dei dati transitori che si applica all'oggetto o se non è stata applicata alcuna politica del ciclo di vita al contenitore, il valore di questo campo è`null`. Questo campo si applica solo alle seguenti operazioni:`PutObject`,`GetObject`,`DescribeObject`,`eDeleteObject`.

HTTPStatus

Il codice di stato HTTP numerico della risposta.

Operazioni

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

Percorso

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 maniera univoca 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 che sono MediaStore stati necessari per elaborare la richiesta. 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 report vengono consegnati entro qualche ora 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 AWS Management Console) 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 iAWS servizi e rispondere automaticamente a eventi di sistema, come i problemi relativi alla disponibilità delle applicazioni o le 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 forniscono notifiche di CloudWatch eventi a Events in pochi secondi, ma a volte può essere necessario 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:

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

Alcuni esempi di utilizzo di CloudWatch Events con AWS Elemental MediaStore includono i seguenti:

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

Per ulteriori informazioni, consulta la Guida per l'utente di Amazon CloudWatch Events.

Argomenti

- Evento di modifica dello stato MediaStore dell'oggetto AWS Elemental
- Evento di modifica dello stato MediaStore del contenitore AWS Elemental

## Evento di modifica dello stato MediaStore dell'oggetto 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 di dati transitoria non emettono un CloudWatch evento quando scadono.

Per informazioni sulla sottoscrizione 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 MediaStore del contenitore AWS Elemental

Questo evento viene pubblicato quando lo stato di un container cambia (quando il container è stato aggiunto o eliminato). Per informazioni sulla sottoscrizione 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 MediaStore Elemental con le CloudWatch metriche di Amazon

Puoi monitorare AWS Elemental MediaStore utilizzando CloudWatch, che raccoglie dati grezzi e li elabora in metriche leggibili. CloudWatchle statistiche di vengono conservate per un periodo di 15 mesi, per consentire l'accesso alle informazioni storiche e ottenere una prospettiva migliore sull'esecuzione del servizio o dell'applicazione 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 Guida per CloudWatch l'utente di Amazon.

Per AWS Elemental MediaStore, potresti voler controllare`BytesDownloaded` e inviare un'email a te stesso quando quella metrica raggiunge una determinata soglia.

Come visualizzare i parametri utilizzando la CloudWatch console

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. Accedere aAWS Management Console e aprire la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, selezionare Parametri.
3. In Tutte le metriche, scegli loMediaStore spazio dei nomi AWS/.
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.

Visualizzazione dei parametri usando AWS CLI

- Al prompt dei comandi, utilizza il comando seguente:

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

## MediaStore Metriche AWS Elemental

La tabella seguente elenca le metriche a cui AWS Elemental MediaStore invia CloudWatch.

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

| Parametro | Descrizione |
| --- | --- |
| `RequestCount` | Il numero totale di richieste HTTP effettuate a un container MediaStore, separate dal tipo di operazione (`Put`, `Get`, `Delete`, `Describe`, `List`).<br><br>Unità: numero<br><br>Dimensioni valide:<br><br>• Nome del container<br>• Nome del gruppo di oggetti<br>• Tipo richiesta<br><br>Statistiche valide: somma |
| `4xxErrorCount` | Il numero di richieste HTTP effettuate MediaStore ha provocato un errore 4xx.<br><br>Unità: numero<br><br>Dimensioni valide:<br><br>• Nome del container<br>• Nome del gruppo di oggetti<br>• Tipo richiesta<br><br>Statistiche valide: somma |

| Parametro | Descrizione |
| --- | --- |
| `5xxErrorCount` | Il numero di richieste HTTP effettuate MediaStore ha provocato un errore 5xx.<br><br>Unità: numero<br><br>Dimensioni valide:<br><br>• Nome del container<br>• Nome del gruppo di oggetti<br>• Tipo richiesta<br><br>Statistiche valide: somma |
| `BytesUploaded` | Il numero di byte caricati per le richieste effettuate a un container MediaStore in cui la richiesta include un corpo.<br><br>Unità: byte<br><br>Dimensioni valide:<br><br>• Nome del container<br>• Nome del gruppo di oggetti<br><br>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 |

| Parametro | Descrizione |
| --- | --- |
| `BytesDownloaded` | Il numero di byte scaricati per le richieste effettuate a un container MediaStore in cui la risposta include un corpo.<br><br>Unità: byte<br><br>Dimensioni valide:<br><br>• Nome del container<br>• Nome del gruppo di oggetti<br><br>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 si MediaStore riceve la richiesta al momento in cui viene inviato 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.<br><br>Unità: millisecondi<br><br>Dimensioni valide:<br><br>• Nome del container<br>• Nome del gruppo di oggetti<br>• Tipo richiesta<br><br>Statistiche valide: media, minimo (come P0,0), massimo (come p100), qualsiasi percentile tra p0,0 e p100 |

| Parametro | Descrizione |
| --- | --- |
| `TurnaroundTime` | Il numero di millisecondi che sono MediaStore stati necessari per elaborare la richiesta. Questo valore viene misurato dal momento in cui si MediaStore riceve l'ultimo byte della richiesta al momento in cui viene inviato il primo byte della risposta.<br><br>Unità: millisecondi<br><br>Dimensioni valide:<br><br>• Nome del container<br>• Nome del gruppo di oggetti<br>• Tipo richiesta<br><br>Statistiche valide: media, minimo (come P0,0), massimo (come p100), qualsiasi percentile tra p0,0 e p100 |
| `ThrottleCount` | Il numero di richieste HTTP effettuate a MediaStore tale scopo è stato limitato.<br><br>Unità: numero<br><br>Dimensioni valide:<br><br>• Nome del container<br>• Nome del gruppo di oggetti<br>• Tipo richiesta<br><br>Statistiche valide: somma |

# Etichettatura delle risorse AWS MediaStore Elemental

Un tag è un'etichetta di attributi personalizzata assegnata dall'utente o da AWS 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. Analogamente alle chiavi dei tag, i valori dei tag prevedono una distinzione tra lettere maiuscole e minuscole.

I tag consentono di eseguire le seguenti operazioni:

- Identificare e organizzare le risorse AWS. 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 lo stesso tag a un contenitore AWS MediaStore *`Elemental`* che assegni a un input. AWS Elemental MediaLive
- Tenere traccia dei costi AWS. Questi tag vengono attivati nel pannello di controllo di AWS Billing and Cost Management. 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 nella codifica del MediaStore supporto AWS Elemental:

- *`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). Le risorse Amazon EC2 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 prefisso `aws:` non può essere utilizzato con i tag; è riservato per l'uso in AWS. 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.

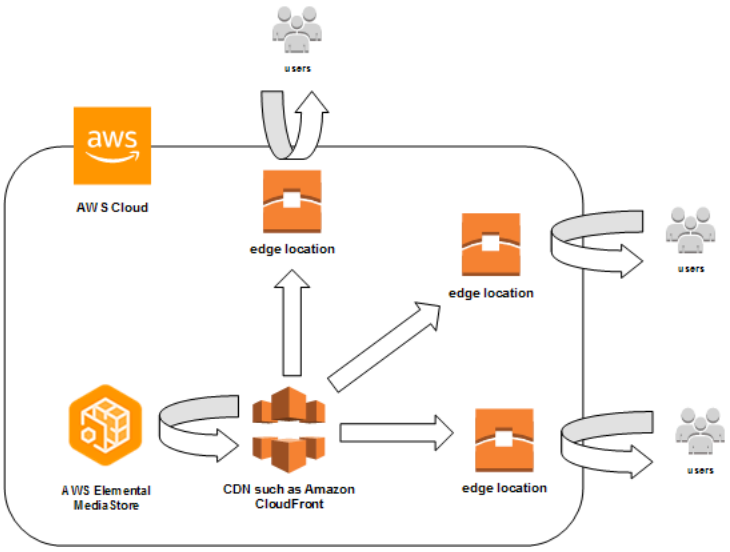
## Gestione dei tag

I tag sono formati dalle proprietà `Key` e `Value` in una risorsa. Puoi 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 delle reti di distribuzione di contenuti (CDN)

Puoi utilizzare una rete di distribuzione di contenuti (CDN) come Amazon CloudFront per distribuire i contenuti archiviati in AWS Elemental MediaStore. 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 il contenuto non si trova attualmente in quella posizione periferica, il CDN lo recupera dall'origine (ad esempio il MediaStore contenitore) e lo distribuisce all'utente.




Argomenti

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

# Consentire CloudFront ad Amazon di accedere al tuo MediaStore contenitore AWS Elemental

Puoi usare Amazon CloudFront per distribuire i contenuti archiviati in un contenitore in AWS Elemental MediaStore. Questa operazione può essere eseguita in uno dei seguenti modi:

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

- Utilizzo di segreti condivisi- Utilizzate questa opzione seRegione 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 MediaStore le origini di AWS Elemental con una maggiore sicurezza. È possibile abilitare AWSSignature 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 console, API, SDK o CLI e non sono previsti costi aggiuntivi per il suo utilizzo.

Per ulteriori informazioni sull'utilizzo della funzionalità OAC con MediaStore, consulta Restricting access to a MediaStore origin nella Amazon CloudFront Developer Guide.

## Utilizzo di segreti condivisi

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

> **Note**
>
> Ti consigliamo di utilizzare la funzione OAC se laRegione AWS supporta. Le seguenti procedure richiedono la configurazione MediaStore e l' CloudFront uso di segreti condivisi per limitare l'accesso ai MediaStore contenitori. Per seguire le migliori pratiche di sicurezza, questa configurazione manuale richiede una rotazione periodica dei segreti. Con OAC on MediaStore origin, puoi chiedere di firmare le richieste utilizzando SigV4 e inoltrarle a esse MediaStore per la corrispondenza delle firme, eliminando la necessità di utilizzare e ruotare i segreti. CloudFront Ciò garantisce che le richieste vengano verificate automaticamente prima che i contenuti multimediali vengano forniti, rendendo la distribuzione dei contenuti multimediali CloudFront più semplice MediaStore e 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 Politica relativa ai container, allega una politica che garantisca ad Amazon l'accesso in lettura o un livello superiore CloudFront.

   Example

   La seguente politica di esempio, simile alla politica di esempio per l'accesso alla lettura pubblica tramite HTTPS, soddisfa questi requisiti perché consente`GetObject` l'invio di`DescribeObject` comandi da parte di 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CloudFrontRead",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "mediastore:GetObject",
        "mediastore:DescribeObject"
      ],
      "Resource": "arn:aws:mediastore:<region>:<owner acct
 number>:container/<container name>/*",
      "Condition": {
        "StringEquals": {
          "aws:Referer": "<secretValue>"
        },
        "Bool": {
          "aws:SecureTransport": "true"
        }
      }
    }
 ]}
```

4. 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). In CloudFront, l'endpoint dei dati viene definito nome di dominio di origine.
   - La struttura della cartella nel container in cui gli oggetti vengono archiviati. Nel CloudFront, questo viene chiamato percorso di origine. Questa impostazione è facoltativa. Per ulteriori informazioni sui percorsi di origine, consulta la Amazon CloudFront Developer Guide.
6. In CloudFront, crea una distribuzione configurata per fornire contenuti da AWS Elemental MediaStore. Saranno necessarie le informazioni raccolte nella fase precedente.

Dopo aver allegato la policy ai MediaStore contenitori, è necessario CloudFront configurare 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. Aprire la CloudFront console.
2. Nella pagina Origini, scegli la tua MediaStore origine.
3. Scegliere Edit (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 implementate.

# Interazione MediaStore di AWS 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 (CDN) come Amazon CloudFront.

Quando un utente finale o una rete CDN recupera un oggetto da MediaStore, il servizio restituisce le 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 CDN e i browser Web conformi agli standard utilizzano questo tag come chiave con cui memorizzare nella cache l'oggetto. MediaStore genera automaticamente un`ETag` per ogni oggetto quando viene caricato. Puoi visualizzare i dettagli di un oggetto per determinarne il valore ETag.
- **`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 lo possiede già.

## Richieste condizionali alle CDN

Le CDN che forniscono contenuti per conto di MediaStore possono elaborare le richieste condizionali restituendole`304 Not Modified`, come descritto nella sezione 4.1 di RFC7232. 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.

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

# Utilizzo di questo servizio con un AWS SDK

AWS i kit di sviluppo software (SDKs) sono disponibili per molti linguaggi di programmazione popolari. Ciascuno di essi SDK fornisceAPI, esempi di codice e documentazione che semplificano agli sviluppatori la creazione di applicazioni nel linguaggio preferito.

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

**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 MediaStore con un AWS kit di sviluppo software (SDK).

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto nei relativi scenari.

Per un elenco completo di AWS SDKguide per sviluppatori ed esempi di codice, vediUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Esempi di codice

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

# Esempi di base per MediaStore l'utilizzo AWS SDKs

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

Esempi

- Azioni per l'utilizzo MediaStore AWS SDKs
  - Utilizzare CreateContainer con un AWS SDKo CLI
  - Utilizzare DeleteContainer con un AWS SDKo CLI
  - Utilizzare DeleteObject con un AWS SDKo CLI

- Utilizzare DescribeContainer con un AWS SDKo CLI
- Utilizzare GetObject con un AWS SDKo CLI
- Utilizzare ListContainers con un AWS SDKo CLI
- Utilizzare PutObject con un AWS SDKo 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 è possibile trovare 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 il AWS Elemental MediaStore APIRiferimento.

Esempi

- Utilizzare CreateContainer con un AWS SDKo CLI
- Utilizzare DeleteContainer con un AWS SDKo CLI
- Utilizzare DeleteObject con un AWS SDKo CLI
- Utilizzare DescribeContainer con un AWS SDKo CLI
- Utilizzare GetObject con un AWS SDKo CLI
- Utilizzare ListContainers con un AWS SDKo CLI
- Utilizzare PutObject con un AWS SDKo CLI

### Utilizzare **`CreateContainer`** con un AWS SDKo CLI

I seguenti esempi di codice 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, vedere Creazione di un contenitore in AWS Guida MediaStore utente elementare.

- Per API i dettagli, vedere CreateContainerin AWS CLI Riferimento ai comandi.

Java

SDKper Java 2.x

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri come configurare ed eseguire in AWS Repository di esempi di codice.

```
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 API i dettagli, vedere CreateContainerin AWS SDK for Java 2.x APIRiferimento.

Per un elenco completo di AWS SDKguide per sviluppatori ed esempi di codice, vediUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

## Utilizzare **`DeleteContainer`** con un AWS SDKo CLI

I seguenti esempi di codice 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, vedere Eliminazione di un contenitore nel AWS Guida MediaStore utente elementare.

- Per API i dettagli, vedere DeleteContainerin AWS CLI Riferimento ai comandi.

Java

SDKper Java 2.x

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri come configurare ed eseguire in AWS Repository di esempi di codice.

```
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 API i dettagli, vedere DeleteContainerin AWS SDK for Java 2.x APIRiferimento.

Per un elenco completo di AWS SDKguide per sviluppatori ed esempi di codice, vediUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

## Utilizzare **`DeleteObject`** con un AWS SDKo CLI

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

Java

SDKper Java 2.x

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri come configurare ed eseguire in AWS Repository di esempi di codice.

```
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 API i dettagli, vedere DeleteObjectin AWS SDK for Java 2.x APIRiferimento.

Per un elenco completo di AWS SDKguide per sviluppatori ed esempi di codice, vediUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

## Utilizzare **`DescribeContainer`** con un AWS SDKo CLI

I seguenti esempi di codice 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, vedere Visualizzazione dei dettagli di un contenitore nella AWS Guida MediaStore utente elementare.

- Per API i dettagli, vedere DescribeContainerin AWS CLI Riferimento ai comandi.

Java

SDKper Java 2.x

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri come configurare ed eseguire in AWS Repository di esempi di codice.

```
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 API i dettagli, vedere DescribeContainerin AWS SDK for Java 2.x APIRiferimento.

Per un elenco completo di AWS SDKguide per sviluppatori ed esempi di codice, vediUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

## Utilizzare **`GetObject`** con un AWS SDKo CLI

I seguenti esempi di codice 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 sull'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, vedete Scaricamento di un oggetto nel AWS Guida MediaStore utente elementare.

- Per API i dettagli, vedere GetObjectin AWS CLI Riferimento ai comandi.

Java

SDKper Java 2.x

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri come configurare ed eseguire in AWS Repository di esempi di codice.

```
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 API i dettagli, vedere GetObjectin AWS SDK for Java 2.x APIRiferimento.

Per un elenco completo di AWS SDKguide per sviluppatori ed esempi di codice, vediUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

## Utilizzare **`ListContainers`** con un AWS SDKo CLI

I seguenti esempi di codice 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://fffggghhhiiijj.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, vedere Visualizzazione di un elenco di contenitori nella AWS Guida MediaStore utente elementare.

- Per API i dettagli, vedere ListContainersin AWS CLI Riferimento ai comandi.

Java

SDKper Java 2.x

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri come configurare ed eseguire in AWS Repository di esempi di codice.

```
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 API i dettagli, vedere ListContainersin AWS SDK for Java 2.x APIRiferimento.

Per un elenco completo di AWS SDKguide per sviluppatori ed esempi di codice, vediUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

## Utilizzare **`PutObject`** con un AWS SDKo CLI

I seguenti esempi di codice 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, vedere Caricamento di un oggetto nella AWS Guida MediaStore utente elementare.

- Per API i dettagli, vedere PutObjectin AWS CLI Riferimento ai comandi.

Java

SDKper Java 2.x

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri come configurare ed eseguire in AWS Repository di esempi di codice.

```
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 API i dettagli, vedere PutObjectin AWS SDK for Java 2.x APIRiferimento.

Per un elenco completo di AWS SDKguide per sviluppatori ed esempi di codice, vediUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

# Quote in AWS Elemental MediaStore

La console Service Quotas fornisce le informazioni sulle MediaStore quote di AWS 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 rappresentano il numero massimo di risorse di servizio o operazioni per l'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 tuoi contenitori, per evitare che un contenitore esaurisca 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 throttling.<br><br>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 throttling.<br><br>È possibile richiedere un aumento della quota. |
| Frequenza delle richieste GetObject API per la disponibilità di caricamento standard | 1.000 | Il numero massimo di richieste di operazioni che puoi effettuare al secondo. Ulteriori richieste verranno sottoposte a throttling.<br><br>È possibile richiedere un aumento della quota. |
| Frequenza delle richieste GetObject API per la disponibilità di caricamento in streaming | 25 | Il numero massimo di richieste di operazioni che puoi effettuare al secondo. Ulteriori richieste verranno sottoposte a throttling.<br><br>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 throttling.<br><br>È possibile richiedere un aumento della quota. |

| Operazione o risorsa | Quota predefinita | Commenti |
| --- | --- | --- |
| Frequenza delle richieste PutObject API per la codifica di trasferimento chunked (nota anche come disponibilità di caricamento in streaming) | 10 | Il numero massimo di richieste di operazioni che puoi effettuare al secondo. Ulteriori richieste verranno sottoposte a throttling.<br><br>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 throttling.<br><br>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 MediaStore relative ad AWS Elemental

La tabella seguente elenca le risorse correlate che si riveleranno utili durante l'utilizzo di AWS Elemental MediaStore.

- Corsi e seminari: collegamenti a corsi basati su ruoli e di specializzazione nonché a corsi gestiti dall'utente per affinareAWS le proprie competenze e acquisire esperienza pratica.
- AWSCentro per sviluppatori: esplora i tutorial, scarica gli strumenti e scopri di più sugli eventi per gliAWS sviluppatori.
- AWSStrumenti per sviluppatori: collegamenti a strumenti per sviluppatori, SDK, kit di strumenti IDE e strumenti a riga di comando per lo sviluppo e la gestione delleAWS applicazioni.
- Centro risorse per iniziare: scopri come configurare il tuoAccount AWS, entrare a far parte dellaAWS community e lanciare la tua prima applicazione.
- Esercitazioni pratiche: scopri le step-by-step esercitazioni per avviare la tua prima applicazione suAWS.
- AWSWhitepaper: collegamenti a un elenco completo diAWS whitepaper tecnici, relativi ad argomenti come l'architettura, la sicurezza e l'economia, creati daAWS Solution Architect o da altri esperti tecnici.
- AWS SupportCentro : il centro in cui creare e gestire i tuoi casi AWS Support. Include inoltre link ad altre risorse utili, quali forum, domande frequenti di tipo tecnico, stato d'integrità del servizio e AWS Trusted Advisor.
- AWS Support- La pagina Web principale che include le informazioni suAWS Support one-on-one, un canale di assistenza rapida che aiuta 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 di utilizzo del sito: informazioni dettagliate sul copyright e i marchi, l'account, la licenza, l'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 abbonarti a un feed RSS.

| Modifica | Descrizione | Data |
| --- | --- | --- |
| Miglioramento dell'Origin Access Control (OAC) | Aggiunta di informazioni su come utilizzare OAC con AWS Elemental MediaStore. | 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 relative ai dati transitori 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 le CloudWatch metriche di Amazon | Puoi impostare una politica di metrica per stabilire a quali metriche MediaStore inviare le metriche CloudWatch. | 30 marzo 2020 |
| Jolly 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 |
| Criteri relativi 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 |
| Supporto AWS CloudFormation | Puoi ora utilizzare un modello AWS CloudFormation per creare un container automaticamente. Il modello AWS 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 di caricamento 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 di oggetti in blocchi | 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 che sono effettuate agli oggetti situate in un contenitore. | 25 febbraio 2019 |
| Criteri relativi 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 |

| | | |
|---|---|---|
| Integrazione di AWS CloudTrail | 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 funzionalità CORS (Cross-Origin Resource Sharing, condivisione delle risorse multiorigine), che consente alle applicazioni Web dei clienti caricate in un dominio possono interagire con le risorse situate in un dominio differente. | 7 febbraio 2018 |
| Nuovo servizio e guida | Questa è la versione iniziale del servizio di origine e archiviazione video, AWS Elemental MediaStore, e della AWS Elemental MediaStore User Guide. | 27 Novembre 2017 |

Note

- I ServiziAWS multimediali non sono progettati o destinati all'uso con applicazioni o in situazioni che richiedono prestazioni a prova di guasto, come operazioni di sicurezza personale, sistemi di navigazione o comunicazione, controllo del traffico aereo o macchine

di supporto vitale in cui l'indisponibilità, l'interruzione o il guasto dei servizi potrebbero causare morte, lesioni personali, danni alla proprietà o danni ambientali.

# Glossario AWS

Per la terminologia AWS più recente, consultare il glossario AWS nella documentazione di riferimento per Glossario AWS.

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.