aws

Benutzerhandbuch

AWS Elemental MediaStore

AWS Elemental MediaStore: Benutzerhandbuch

Table of Contents

Was ist AWS Elemental MediaStore?

AWS Elemental MediaStore ist ein Service zur Erstellung und Speicherung von Videos, der die hohe Leistung und sofortige Konsistenz bietet, die für die Live-Erzeugung erforderlich sind. Mit MediaStore können Sie Video-Assets als Objekte in Containern verwalten, um zuverlässige, cloudbasierte Medien-Workflows zu erstellen.

Um den Service zu nutzen, laden Sie Ihre Objekte aus einer Quelle, z. B. von einem Encoder oder Datenfeed, in einen Container hoch, den Sie in MediaStore erstellen.

MediaStore ist eine hervorragende Wahl für das Speichern fragmentierter Videodateien, wenn Sie hohe Konsistenz, Lese- und Schreibvorgänge mit geringer Latenz und die Fähigkeit benötigen, große Mengen gleichzeitiger Anfragen zu verarbeiten. Wenn Sie keine Live-Streaming-Videos bereitstellen, sollten Sie stattdessen Amazon Simple Storage Service (Amazon S3) verwenden.

Themen

MediaStore Konzepte und Terminologie von AWS Elemental

ARN

Ein Amazon-Ressourcenname.

Fließtext

Die Daten, die in ein Objekt hochgeladen werden sollen.

(Byte)-Bereich

Eine Untermenge der Objektdaten, die angesprochen werden. Weitere Informationen finden Sie unter Bereich in der HTTP-Spezifikation.

Container

Ein Namespace, der Objekte enthält. Ein Container hat einen Endpunkt, den Sie zum Schreiben und Abrufen von Objekten und zum Anfügen von Zugriffsrichtlinien verwenden können.

Endpunkt

Ein Einstiegspunkt zum MediaStore Service, der als HTTPS-Root-URL angegeben wird.

ETag

Ein Entity-Tag, das ist ein Hash der Objektdaten.

Ordner

Eine Abschnitt eines Containers. Ein Ordner kann Objekte und andere Ordner enthalten.

Item

Ein Begriff für Objekte und Ordner.

Objekt

Ein Asset, ähnlich einem Amazon S3 S3-Objekt. Objekte sind die grundlegenden Einheiten, die in MediaStore gespeichert sind. Der Service akzeptiert alle Dateitypen.

Bereitstellungsservice

MediaStore wird als Ursprungsdienst betrachtet, da er die Vertriebsstelle für die Bereitstellung von Medieninhalten ist.

Pfad

Eine eindeutige ID für ein Objekt oder einen Ordner, die deren Position im Container angibt.

Teil

Eine Teilmenge der Daten (Block) eines Objekts.

Richtlinie

Eine IAM-Richtlinie.

Ressource

Eine Entität in AWS, mit der Sie arbeiten können. Jeder AWS-Ressource ist ein Amazon-Ressourcenname (ARN) zugeordnet, der als eindeutige Kennung fungiert. In MediaStore, das ist die Ressource und ihr ARN-Format:

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

Zugehörige Services

- Amazon CloudFront ist ein globaler Content Delivery Network (CDN) -Service, der Daten und Videos sicher an Ihre Zuschauer liefert. Verwenden Sie CloudFront, um Inhalte mit der bestmöglichen Leistung bereitzustellen. Weitere Informationen finden Sie im Amazon CloudFront Developer Guide.
- AWS CloudFormation ist ein Service, mit dem Sie AWS-Ressourcen modellieren und einrichten können. Sie erstellen eine Vorlage, die alle gewünschten AWS Ressourcen (wie MediaStore Container) beschreibt und sich um die Bereitstellung und Konfiguration dieser Ressourcen für Sie AWS CloudFormation kümmert. Sie müssen die AWS-Ressourcen nicht einzeln erstellen und konfigurieren und herausfinden, welche Abhängigkeiten bestehen. AWS CloudFormation kümmert sich um alles. Weitere Informationen finden Sie im AWS CloudFormation-Benutzerhandbuch.
- AWS CloudTrailist ein Service, mit dem Sie die CloudTrail API-Aufrufe für Ihr Konto, einschließlich der Aufrufe der AWS-ManagementkonsoleAWS CLI, und anderer Services überwachen können. Weitere Informationen finden Sie im AWS CloudTrail-Benutzerhandbuch.
- Amazon CloudWatch ist ein Überwachungsservice für AWS Cloud-Ressourcen und die Anwendungen, auf denen Sie laufenAWS. Verwenden Sie CloudWatch Ereignisse, um Statusänderungen von Containern und Objekten in zu verfolgen MediaStore. Weitere Informationen finden Sie in der CloudWatch Amazon-Dokumentation.
- AWS Identity and Access Management (IAM) ist ein Webservice, mit dem Sie auf sichere Weise den Zugriff auf AWS-Ressourcen für Ihre Benutzer steuern können. Sie verwenden IAM, um zu steuern, wer Ihre AWS-Ressourcen verwenden kann (Authentifizierung) und welche Ressourcen von Benutzern auf welche Weise verwendet werden können (Autorisierung). Weitere Informationen finden Sie unter Einrichtung von AWS Elemental MediaStore.
- Amazon Simple Storage Service (Amazon S3) ist ein Objektspeicher, der entwickelt wurde, um beliebige Datenmengen von überall zu speichern und abzurufen. Weitere Informationen finden Sie in der Amazon S3-Dokumentation.

Zugreifen auf AWS Elemental MediaStore

Sie können MediaStore mit einer der folgenden Methoden zugreifen:

- AWS-Managementkonsole — Die Verfahren in diesem Handbuch erläutern, wie Sie die AWS-Managementkonsole zur Ausführung von Aufgaben für verwenden MediaStore. So greifen MediaStore Sie über die Konsole zu:

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

- AWS Command Line Interface— Weitere Informationen finden Sie im AWS Command Line InterfaceBenutzerhandbuch. So greifen MediaStore Sie über den CLI-Endpunkt zu:

  ```
  aws mediastore
  ```

- MediaStore API — Wenn Sie eine Programmiersprache verwenden, für die kein SDK verfügbar ist, finden Sie in der AWS Elemental MediaStoreAPI-Referenz Informationen zu API-Aktionen und zum Stellen von API-Anfragen. So greifen MediaStore Sie über den REST-API-Endpunkt zu:

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

- AWS-SDKs: Wenn Sie eine Programmiersprache verwenden, für die AWS ein SDK anbietet, können Sie ein SDK verwenden, um auf MediaStore zuzugreifen. SDKs vereinfachen die Authentifizierung, lassen sich leicht in die Entwicklungsumgebung integrieren und bieten einen einfachen Zugriff auf MediaStore -Befehle. Weitere Informationen finden Sie unter Tools für Amazon Web Services.
- AWS-Tools für Windows PowerShell — Weitere Informationen finden Sie im AWS Tools for Windows PowerShellBenutzerhandbuch.

Preise für AWS Elemental MediaStore

Wie bei anderen AWS Produkten gibt es keine Verträge oder Mindestverpflichtungen für die Nutzung MediaStore. Ihnen wird eine Gebühr pro GB für die Aufnahme von Inhalten in den Service und eine monatliche Gebühr pro GB für Inhalte, die Sie im Service speichern, in Rechnung gestellt. Weitere Informationen finden Sie unter AWS Elemental MediaStore Pricing.

Regionen und Endpunkte für AWS Elemental MediaStore

Um die Datenlatenz in Ihren Anwendungen zu reduzieren, MediaStore bietet es einen regionalen Endpunkt, an dem Sie Ihre Anfrage stellen können:

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

Eine vollständige Liste der AWS-Regionen, in denen sie verfügbar MediaStore ist, finden Sie unter AWS Elemental MediaStore Endpoints and Quotas in der AWS General Reference.

Einrichtung von AWS Elemental MediaStore

Dieser Abschnitt führt Sie durch die Schritte, die erforderlich sind, um Benutzer für den Zugriff auf AWS Elemental MediaStore zu konfigurieren. Hintergrundinformationen und zusätzliche Informationen zur Identitäts- und Zugriffsverwaltung für finden Sie MediaStore unterIdentity and Access Management für AWS Elemental MediaStore.

Führen Sie die folgenden Schritte aus MediaStore, um mit der Nutzung von AWS Elemental zu beginnen.

Themen

Melden Sie sich für eine an AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/signup.
2. Folgen Sie den Online-Anweisungen.

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Aus Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich ist.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu https://aws.amazon.com/ auf und klicken Sie auf Mein Konto.

Erstellen Sie einen Benutzer mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS Management Consoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung Benutzerhandbuch zu.

2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen Sie einen Benutzer mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.

2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis im AWS IAM Identity Center Benutzerhandbuch.

Melden Sie sich als Benutzer mit Administratorzugriff an

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Zugriffsportal.

Weisen Sie weiteren Benutzern Zugriff zu

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

   Anweisungen finden Sie im Benutzerhandbuch unter Einen Berechtigungssatz erstellen.AWS IAM Identity Center
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Anweisungen finden Sie im AWS IAM Identity Center Benutzerhandbuch unter Gruppen hinzufügen.

Erste Schritte mit AWS Elemental MediaStore

Dieses Getting Started Tutorial zeigt Ihnen, wie Sie AWS Elemental verwenden MediaStore , um einen Container zu erstellen und ein Objekt hochzuladen.

Themen

Schritt 1: Zugriff auf AWS Elemental MediaStore

Nachdem Sie Ihr AWS-Konto eingerichtet und Benutzer und Rollen erstellt haben, melden Sie sich bei der Konsole für AWS Elemental an MediaStore.

So greifen Sie auf AWS Elemental zu MediaStore

- Melden Sie sich bei der anAWS Management Console und öffnen Sie die MediaStore Konsole unter https://console.aws.amazon.com/mediastore/.

> **Note**
>
> Sie können sich mit beliebigen IAM-Anmeldeinformationen anmelden, die Sie für dieses Konto erstellt haben. Weitere Informationen über das Erstellen IAM-Anmeldeinformationen finden Sie unter Einrichtung von AWS Elemental MediaStore.

Schritt 2: Erstellen eines Containers

Sie verwenden Container in AWS Elemental MediaStore , um Ihre Ordner und Objekte zu speichern. Sie können Container verwenden, um verwandte Objekte auf ähnliche Weise zu gruppieren, wie Sie ein Verzeichnis verwenden, um Dateien in einem Dateisystem zu gruppieren. Es entstehen Ihnen keine Kosten, wenn Sie Container erstellen; Gebühren fallen nur dann an, wenn Sie ein Objekt in einen Container hochladen.

Erstellen eines Containers

1. Wählen Sie auf der Seite Containers (Container) die Option Create Container (Container erstellen) aus.
2. Geben Sie für Container name (Containername) den Namen für Ihren Container ein. Weitere Informationen finden Sie unter Regeln für Containernamen.
3. Wählen Sie Container erstellen. AWS Elemental MediaStore fügt den neuen Container einer Liste von Containern hinzu. Anfänglich ist der Status des Containers Creating (Wird erstellt), dann wechselt er zu Active (Aktiv).

Schritt 3: Hochladen eines Objekts

Sie können Objekte (jeweils bis zu 25 MB) in einen Container oder in einen Ordner innerhalb eines Containers hochladen. Um ein Objekt in einen Ordner hochzuladen, geben Sie den Pfad zum Ordner an. Wenn der Ordner bereits existiert, MediaStore speichert AWS Elemental das Objekt in dem Ordner. Wenn der Ordner nicht vorhanden, legt der Service ihn an und speichert das Objekt in dem Ordner.

> **Note**
>
> Objektdateinamen dürfen nur Buchstaben, Ziffern, Punkte (.), Unterstriche (_), Tilden (~) und Bindestriche (-) enthalten.

So laden Sie ein Objekt hoch

1. Wählen Sie auf der Seite Containers (Container) den Namen des Containers, den Sie soeben erstellt haben. Die Detailseite für den Container wird angezeigt.
2. Wählen Sie Upload object (Objekt hochladen).
3. Geben Sie für Target path (Zielpfad) einen Pfad für die Ordner ein. Zum Beispiel `premium/canada`. Wenn einer der Ordner im Pfad noch nicht existiert, MediaStore erstellt AWS Elemental ihn automatisch.
4. Wählen Sie für Object (Objekt) Browse (Durchsuchen).
5. Navigieren Sie zum entsprechenden Ordner und wählen Sie ein Objekt zum Hochladen aus.
6. Wählen Sie Open (Öffnen) und anschließend Upload (Hochladen).

Schritt 4: Zugreifen auf ein Objekt

Sie können Ihre Objekte auf einen bestimmten Endpunkt herunterladen.

1. Wählen Sie auf der Seite Containers (Container) den Namen des Containers, der das Objekt enthält, das Sie herunterladen möchten.
2. Wenn sich das Objekt, das Sie herunterladen möchten, in einem Unterordner befindet, wählen Sie so lange Ordnernamen aus, bis Sie das Objekt sehen.
3. Wählen Sie den Namen des Objekts.
4. Wählen Sie auf der Detailseite für das Objekt die Option Download (Herunterladen).

Container in AWS ElementalMediaStore

Sie verwenden Container in MediaStore zum Speichern Ihrer Ordner und Objekte. Verwandte Objekte können in Containern gruppiert werden, auf ähnliche Weise, wie Sie ein Verzeichnis verwenden, um Dateien in einem Dateisystem zu gruppieren. Es entstehen Ihnen keine Kosten, wenn Sie Container erstellen; Gebühren fallen nur dann an, wenn Sie ein Objekt in einen Container hochladen. Weitere Informationen zu Gebühren finden Sie unterAWS ElementalMediaStorePreiseaus.

Themen

Regeln für Containernamen

Wenn Sie einen Namen für Ihren Container wählen, beachten Sie Folgendes:

- Der Name muss innerhalb des aktuellen Kontos für die aktuelle AWS-Region eindeutig sein.
- Der Name kann Großbuchstaben, Kleinbuchstaben, Ziffern und Unterstriche (_) enthalten.
- Der Name muss zwischen 1 und 255 Zeichen lang sein.
- Bei den Namen muss die Groß- und Kleinschreibung beachtet werden. Sie können beispielsweise einen Container mit dem Namen `myContainer` und einen Ordner mit dem Namen `mycontainer` verwenden, weil diese Namen eindeutig sind.
- Ein Container kann nach dem Erstellen nicht mehr umbenannt werden.

Erstellen eines Containers

Sie können bis zu 100 Container für jedes AWS-Konto erstellen. Sie können eine beliebige Anzahl von Ordnern erstellen, solange sie nicht mehr als 10 Ebenen innerhalb eines Containers verschachtelt sind. Darüber hinaus können Sie beliebig viele Objekte in jeden Container hochladen.

> **Tip**
>
> Sie können einen Container auch automatisch mithilfe einer AWS CloudFormation-Vorlage erstellen. Die AWS CloudFormation-Vorlage verwaltet Daten für fünf API-Aktionen: Erstellung eines Containers, Einstellung der Zugriffsprotokollierung, Aktualisierung der Standard-Containerrichtlinie, Hinzufügen einer CORS-Richtlinie (Cross-Origin Resource Sharing) und Hinzufügen einer Objektlebenszyklusrichtlinie. Weitere Informationen finden Sie im AWS CloudFormation-Benutzerhandbuch.

Einen Container erstellen (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) die Option Create Container (Container erstellen) aus.
3. Geben Sie für Container den Namen für den Container ein. Weitere Informationen finden Sie unter Regeln für Containernamen .
4. Klicken Sie aufContainer erstellenaus. AWS ElementalMediaStorefügt den neuen Container einer Liste von Containern hinzu. Anfänglich ist der Status des Containers Creating (Wird erstellt), dann wechselt er zu Active (Aktiv).

Erstellen eines Containers (AWS CLI)

- Verwenden Sie in der AWS CLI den Befehl `create-container`:

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

Im folgenden Beispiel finden Sie den Rückgabewert:

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

```
    }
}
```

Anzeigen der Details für einen Container

Details für einen Container sind unter anderem die Container-Richtlinie, Endpunkt, ARN und den Zeitpunkt der Erstellung.

Anzeigen der Details für einen Container (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) den Containernamen aus.

   Die Seite mit den Containerdetails wird angezeigt. Diese Seite ist in zwei Abschnitte unterteilt:

   - Den Abschnitt Objects (Objekte), der die Objekte und Ordner im Container auflistet.
   - Den Container-Richtlinienabschnitt, der die ressourcenbasierte Richtlinie zeigt, die diesem Container zugeordnet ist. Weitere Informationen über Ressourcenrichtlinien finden Sie unter Containerrichtlinien.

Anzeigen der Details für einen Container (AWS CLI)

- Verwenden Sie in der AWS CLI den Befehl `describe-container`:

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

  Im folgenden Beispiel finden Sie den Rückgabewert:

  ```
  {
      "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"
  ```

```
    }
}
```

Anzeigen einer Liste von Containern

Sie können eine Liste aller Container anzeigen, die mit Ihrem Konto verknüpft sind.

Eine Liste der Container anzeigen (Konsole)

- Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.

  Die Seite Containers (Container) wird angezeigt und listet alle Container auf, die Ihrem Konto zugeordnet sind.

Eine Liste der Container anzeigen (AWS CLI)

- Verwenden Sie in der AWS CLI den Befehl `list-containers`.

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

  Im folgenden Beispiel finden Sie den Rückgabewert:

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

Löschen eines Containers

Sie können einen Container nur löschen, wenn er keine Objekte enthält.

Einen Container löschen (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) die Option links neben dem Containernamen.
3. Wählen Sie Delete (Löschen).

Einen Container löschen (AWS CLI)

- Verwenden Sie in der AWS CLI den Befehl `delete-container`:

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

  Dieser Befehl hat keinen Rückgabewert.

Richtlinien in AWS ElementalMediaStore

Sie können eine oder mehrere der folgenden Richtlinien auf Ihr AWS Elementar anwendenMediaStoreContainer:

- Containerrichtlinie- Legt die Zugriffsrechte auf alle Ordner und Objekte innerhalb des Containers fest. MediaStorelegt eine Standardrichtlinie fest, mit der Benutzer alle ausführen könnenMediaStoreOperationen am Container. Diese Richtlinie legt fest, dass alle Vorgänge über HTTPS ausgeführt werden müssen. Nachdem Sie einen Container erstellt haben, können Sie die Containerrichtlinie bearbeiten.
- Cross-Origin Resource Sharing (CORS) -Richtlinie- Ermöglicht Client-Webanwendungen aus einer Domain, mit Ressourcen in einer anderen Domain zu interagieren. MediaStorelegt keine Standard-CORS-Richtlinie fest.
- Metrikrichtlinie- ErmöglichtMediaStoreum -Metriken an Amazon zu sendenCloudWatchaus. MediaStorelegt keine Standardmetrikrichtlinie fest.
- Objektlebenszyklus-Richtlinie- Steuert, wie lange Objekte in einemMediaStore-Container. MediaStorelegt keine Objektlebenszyklus-Standardrichtlinie fest.

Container-Richtlinien in AWS ElementarMediaStore

Jeder Container hat eine ressourcenbasierte Richtlinie, die die Zugriffsrechte auf alle Ordner und Objekte in diesem Container regelt. Die Standardrichtlinie, die automatisch an alle neuen Container angefügt wird, erlaubt den Zugriff auf alle AWS ElementalMediaStoreOperationen am Container. Sie gibt vor, dass dieser Zugriff die Bedingung hat, dass HTTPS für die Operationen verwendet wird. Nach dem Erstellen eines Containers können Sie die Richtlinie bearbeiten, die an diesen Container angehängt ist.

Sie können auch eine Objektlebenszyklus-Richtlinie angeben, die das Ablaufdatum der Objekte in einem Container regelt. Wenn Objekte das angegebene Höchstalter erreichen, löscht der Service die Objekte aus dem Container.

Themen

- Anzeigen einer Containerrichtlinie
- Bearbeiten einer Containerrichtlinie
- Beispiel-Containerrichtlinien

Anzeigen einer Containerrichtlinie

Sie können die Konsole oder die AWS CLI verwenden, um die ressourcenbasierte Richtlinie eines Containers anzuzeigen.

Anzeige einer Containerrichtlinie (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) den Containernamen aus.

   Die Seite mit den Containerdetails wird angezeigt. Die Richtlinie wird im Abschnitt Container Policy (Containerrichtlinie) angezeigt.

Anzeige einer Containerrichtlinie (AWS CLI)

- Verwenden Sie in der AWS CLI den Befehl `get-container-policy`:

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

  Im folgenden Beispiel finden Sie den Rückgabewert:

  ```
  {
    "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"
  ```

```
          }
        }
      }
    ]
  }
}
```

Bearbeiten einer Containerrichtlinie

Sie können die Berechtigungen in der Standard-Containerrichtlinie bearbeiten oder eine neue Richtlinie erstellen, die die Standardrichtlinie ersetzt. Es dauert bis zu fünf Minuten, bis die neue Richtlinie wirksam wird.

Bearbeiten einer Containerrichtlinie (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) den Containernamen aus.
3. Wählen Sie Edit policy (Richtlinie bearbeiten). Beispiele, die zeigen, wie verschiedene Berechtigungen eingerichtet werden, finden Sie unter the section called “Beispiel-Containerrichtlinien”.
4. Nehmen Sie die entsprechenden Änderungen vor und wählen Sie dann Save (Speichern) aus.

Bearbeiten einer Containerrichtlinie (AWS CLI)

1. Erstellen Sie eine Datei, mit der die Containerrichtlinie definiert wird:

```
{
  "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. Verwenden Sie in der AWS CLI den Befehl `put-container-policy`:

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

Dieser Befehl hat keinen Rückgabewert.

Beispiel-Containerrichtlinien

Die folgenden Beispiele zeigen Containerrichtlinien, die für verschiedene Benutzergruppen erstellt wurden.

Themen

- Beispiel-Containerrichtlinie: Standard
- Beispiel-Containerrichtlinie: Öffentlicher Lesezugriff über HTTPS
- Beispiel-Containerrichtlinie: Öffentlicher Lesezugriff über HTTP oder HTTPS
- Beispiel-Containerrichtlinie: Kontoübergreifender Lesezugriff – HTTP-fähig
- Beispiel-Containerrichtlinie: Kontoübergreifender Lesezugriff über HTTPS
- Beispiel-Containerrichtlinie: Kontoübergreifender Lesezugriff für eine Rolle
- Beispiel-Containerrichtlinie: Kontoübergreifender Vollzugriff für eine Rolle
- Beispiel-Containerrichtlinie: Zugriff auf bestimmte IP-Adressen beschränkt

Beispiel-Containerrichtlinie: Standard

Wenn Sie einen Container erstellen, wird AWS ElementarMediaStorefügt automatisch die folgende ressourcenbasierte Richtlinie hinzu:

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

Die Richtlinie ist in den Service integriert, Sie müssen sie also nicht erstellen. Sie können jedochBearbeiten Sie die Richtlinieauf dem Container, wenn die Berechtigungen in der Standardrichtlinie nicht mit den Berechtigungen übereinstimmen, die Sie für den Container verwenden möchten.

Die Standardrichtlinie, die allen neuen Containern zugewiesen wird, erlaubt den Zugriff auf alle MediaStore-Operationen für den Container. Sie gibt vor, dass dieser Zugriff die Bedingung hat, dass HTTPS für die Operationen verwendet wird.

Beispiel-Containerrichtlinie: Öffentlicher Lesezugriff über HTTPS

Diese Beispielrichtlinie erlaubt Benutzern das Abrufen eines Objekts über eine HTTPS-Anfrage. Dies ermöglicht Lesezugriff für alle Benutzer über eine sichere SSL/TLS-Verbindung: authentifizierte Benutzer und anonyme Benutzer (Benutzer, die nicht angemeldet sind). Die Anweisung hat den Namen `PublicReadOverHttps`. Sie erlaubt Zugriff auf die Operationen `GetObject` und `DescribeObject` sowie auf beliebige Objekte (wie durch den * am Ende des Ressourcenpfads angegeben). Sie gibt vor, dass dieser Zugriff die Bedingung hat, dass HTTPS für die Operationen verwendet wird:

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

Beispiel-Containerrichtlinie: Öffentlicher Lesezugriff über HTTP oder HTTPS

Diese Beispielrichtlinie erlaubt Zugriff auf die Operationen `GetObject` und `DescribeObject` sowie auf beliebige Objekte (wie durch den * am Ende des Ressourcenpfads angegeben). Sie erlaubt allen Benutzern Lesezugriff: allen authentifizierten Benutzern und anonymen Benutzern (Benutzer, die nicht angemeldet sind):

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

Beispiel-Containerrichtlinie: Kontoübergreifender Lesezugriff – HTTP-fähig

Diese Beispielrichtlinie erlaubt Benutzern das Abrufen eines Objekts über eine HTTP-Anfrage. Sie erlaubt diesen Zugriff authentifizierten Benutzern mit kontoübergreifendem Zugriff. Das Objekt muss nicht auf einem Server mit einem SSL/TLS-Zertifikat gehostet sein:

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

Beispiel-Containerrichtlinie: Kontoübergreifender Lesezugriff über HTTPS

Diese Beispielrichtlinie erlaubt den Zugriff auf die`GetObject`und`DescribeObject`Operationen für ein beliebiges Objekt (wie durch den * am Ende des Ressourcenpfads angegeben), das im Besitz des Root-Benutzers der angegebenen ist<other acct number>. Sie gibt vor, dass dieser Zugriff die Bedingung hat, dass HTTPS für die Operationen verwendet wird:

```
{
  "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"
        }
      }
```

```
    }
  ]
}
```

Beispiel-Containerrichtlinie: Kontoübergreifender Lesezugriff für eine Rolle

Die Beispielrichtlinie erlaubt den Zugriff auf die Operationen `GetObject` und `DescribeObject` sowie auf beliebige Objekte (wie durch den * am Ende des Ressourcenpfads angegeben), die im Besitz der angegebenen <Eigentümer-Kontonummer> sind. Sie erlaubt diesen Zugriff jedem Benutzer der <anderen Kontonummer>, wenn dieses Konto die Rolle übernommen hat, die in <Rollenname> angegeben ist:

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

Beispiel-Containerrichtlinie: Kontoübergreifender Vollzugriff für eine Rolle

Diese Beispielrichtlinie erlaubt kontoübergreifenden Zugriff zur Aktualisierung eines beliebigen Objekts im Konto, wenn der Benutzer über HTTP angemeldet ist. Außerdem erlaubt sie kontoübergreifenden Zugriff, um Objekte über HTTP oder HTTPS in einem Konto zu löschen, herunterzuladen und zu beschreiben, das die angegebene Rolle angenommen hat.

- Die erste Anweisung ist `CrossAccountRolePostOverHttps`. Sie erlaubt den Zugriff auf die Operation `PutObject` für ein beliebiges Objekt und erlaubt diesen Zugriff jedem beliebigen Benutzer des angegebenen Kontos, wenn dieses Konto die Rolle übernommen hat, die in <Rollenname> angegeben ist. Sie gibt an, dass dieser Zugriff die Bedingung hat, dass HTTPS für die Operation gefordert wird (diese Bedingung muss immer enthalten sein, wenn Zugriff auf `PutObject` erteilt wird).

Mit anderen Worten, jeder Prinzipal, der kontoübergreifenden Zugriff besitzt, kann auf `PutObject` zugreifen, aber nur über HTTPS.

- Die zweite Anweisung ist `CrossAccountFullAccessExceptPost`. Sie erlaubt Zugriff auf alle Operationen außer `PutObject` für jedes Objekt. Sie erlaubt diesen Zugriff jedem Benutzer des angegebenen Kontos, wenn dieses Konto die Rolle übernommen hat, die in <Rollenname> angegeben ist. Dieser Zugriff hat nicht die Bedingung hat, dass HTTPS für die Operationen gefordert wird.

  Mit anderen Worten, jedes Konto mit kontoübergreifendem Zugriff kann auf `DeleteObject`, `GetObject` usw. (aber nicht `PutObject`) zugreifen, und dies über HTTP oder HTTPS.

  Wenn Sie `PutObject` nicht von der zweiten Anweisung ausschließen, ist die Anweisung nicht gültig (weil Sie HTTPS explizit als Bedingung vorgeben müssen, wenn Sie `PutObject` aufnehmen).

```
{
  "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>/*"
```

```
    }
  ]
}
```

Beispiel-Containerrichtlinie: Zugriff auf bestimmte IP-Adressen beschränkt

Diese Beispielrichtlinie erlaubt den Zugriff auf alle AWS ElementalMediaStore-Operationen für Objekte im angegebenen Container. Die Anfrage muss jedoch aus dem in der Bedingung angegebenen IP-Adressbereich stammen.

Die Bedingung in dieser Anweisung identifiziert den Bereich 198.51.100.* als zulässigen Bereich für Internetprotokoll 4-Adressen (IPv4-Adressen), mit einer Ausnahme: 198.51.100.188.

Der `Condition`-Block verwendet die Bedingungen `IpAddress` und `NotIpAddress` und den Bedingungsschlüssel `aws:SourceIp`, wobei es sich um einen AWS-übergreifenden Bedingungsschlüssel handelt. Die `aws:sourceIp` IPv4-Werte verwenden die CIDR-Standardnotation. Weitere Informationen finden Sie unterBedingungsoperatoren für IP-Adressenim IAM User Guide.

```
{
    "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"
                }
            }
```

```
        }
    ]
}
```

Cross-Origin Resource Sharing (CORS) -Richtlinien in AWS ElementalMediaStore

Cross-Origin Resource Sharing (CORS) bestimmt für Client-Webanwendungen, die in einer Domain geladen sind, eine Möglichkeit zur Interaktion mit Ressourcen in einer anderen Domain. Mit CORS-Unterstützung in AWS ElementarMediaStoreerstellen können Sie umfangreiche clientseitige Webanwendungen mitMediaStoreund gestatten Sie selektiv den Zugang zu IhremMediaStoreRessourcen schätzen.

> **Note**
>
> Wenn Sie Amazon verwendenCloudFrontum Inhalte aus einem Container zu verteilen, der über eine CORS-Richtlinie verfügt, müssen Sie unbedingtkonfigurieren Sie die Distribution für AWS ElementarMediaStore(einschließlich des Schrittes zum Bearbeiten des Cacheverhaltens zum Einrichten von CORS).

Dieser Abschnitt bietet eine Übersicht über CORS. Die Unterthemen beschreiben, wie Sie CORS mit AWS Elemental aktivieren könnenMediaStore-Konsole oder programmgesteuert mit demMediaStoreREST API und die AWS SDKs.

Themen

CORS-Anwendungsfälle

Es folgen typische Beispielszenarien für den Einsatz von CORS:

- Szenario 1: Angenommen, Sie verteilen Live-Streaming-Videos in einem AWS ElementalMediaStore-ContainerLiveVideoaus. Ihre Benutzer laden den Video-Manifest-Endpunkt `http://livevideo.mediastore.ap-southeast-2.amazonaws.com` von einem bestimmten Ursprungsserver wie beispielsweise `www.example.com`. Sie möchten einJavaScript-Videoplayer für den Zugriff auf Videos, die aus diesem -Container stammen, über nicht authentifizierte`GET`und`PUT`anfragen. Ein Browser würde normalerweise blockierenJavaScriptSie können diese Anfragen nicht zulassen, aber Sie können eine CORS-Richtlinie für Ihren -Container festlegen, um diese Anfragen aus`www.example.com`aus.
- Szenario 2: Angenommen, Sie möchten den gleichen Live-Stream wie in Szenario 1 ausMediaStoreContainer, aber Anfragen von jedem Ursprung zulassen möchten. Sie können eine CORS-Richtlinie so konfigurieren, dass Wildcard-(*)-Ursprünge erlaubt sind, sodass Anfragen von jedem beliebigen Ursprung auf das Video zugreifen können.

Hinzufügen einer CORS-Richtlinie zu einem Container

Dieser Abschnitt erklärt, wie Sie einem AWS Elemental eine Cross-Origin Resource Sharing (CORS) -Konfiguration hinzufügenMediaStore-Container. CORS erlaubt Client-Webanwendungen, die in einer Domäne geladen sind, eine Möglichkeit zur Interaktion mit Ressourcen in einer anderen Domäne.

Um Ihren Container so zu konfigurieren, dass er ursprungsübergreifende Anfragen zulässt, fügen Sie dem Container eine CORS-Richtlinie hinzu. Eine CORS-Richtlinie definiert Regeln, die die Ursprünge identifizieren, die den Zugriff auf Ihren Container zulassen, die Operationen (HTTP-Methoden), die für jeden Ursprung unterstützt werden, sowie weitere operationsspezifische Informationen.

Wenn Sie dem Container eine CORS-Richtlinie hinzufügen, gelten die Container-Richtlinien (die die Zugriffsrechte auf den Container regeln) weiterhin.

Hinzufügen einer CORS-Richtlinie (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) den Namen des Containers, für den Sie eine CORS-Richtlinie erstellen möchten.

Die Seite mit den Containerdetails wird angezeigt.

3. Wählen Sie im Abschnitt Container CORS policy (Container-CORS-Richtlinie) die Option Create CORS policy (CORS-Richtlinie erstellen).
4. Fügen Sie die Richtlinie im JSON-Format ein und wählen Sie Save (Speichern).

Hinzufügen einer CORS-Richtlinie (AWS CLI)

1. Erstellen Sie eine Datei, mit der die CORS-Richtlinie definiert wird:

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

2. Verwenden Sie in der AWS CLI den Befehl `put-cors-policy`.

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

Dieser Befehl hat keinen Rückgabewert.

Anzeigen einer CORS-Richtlinie

Cross-Origin Resource Sharing (CORS) bestimmt für Client-Webanwendungen, die in einer Domain geladen sind, eine Möglichkeit zur Interaktion mit Ressourcen in einer anderen Domain.

Anzeige einer CORS-Richtlinie (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) den Namen des Containers, für den Sie eine CORS-Richtlinie anzeigen möchten.

   Die Container-Detailseite wird angezeigt, mit der CORS-Richtlinie im Abschnitt Container CORS Policy (Container--CORS-Richtlinie).

Anzeige einer CORS-Richtlinie (AWS CLI)

- Verwenden Sie in der AWS CLI den Befehl `get-cors-policy`:

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

  Im folgenden Beispiel finden Sie den Rückgabewert:

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

Bearbeiten einer CORS-Richtlinie

Cross-Origin Resource Sharing (CORS) bestimmt für Client-Webanwendungen, die in einer Domain geladen sind, eine Möglichkeit zur Interaktion mit Ressourcen in einer anderen Domain.

Bearbeiten einer CORS-Richtlinie (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) den Namen des Containers, für den Sie eine CORS-Richtlinie bearbeiten möchten.

   Die Seite mit den Containerdetails wird angezeigt.
3. Wählen Sie im Abschnitt Container CORS policy (Container-CORS-Richtlinie) die Option Edit CORS policy (CORS-Richtlinie bearbeiten).
4. Nehmen Sie Ihre Änderungen an der Richtlinie vor und wählen Sie Save (Speichern).

So bearbeiten Sie eine CORS-Richtlinie (AWS CLI)

1. Erstellen Sie eine Datei, mit der die aktualisierte CORS-Richtlinie definiert wird:

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

2. Verwenden Sie in der AWS CLI den Befehl `put-cors-policy`.

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

   Dieser Befehl hat keinen Rückgabewert.

Löschen einer CORS-Richtlinie

Cross-Origin Resource Sharing (CORS) bestimmt für Client-Webanwendungen, die in einer Domain geladen sind, eine Möglichkeit zur Interaktion mit Ressourcen in einer anderen Domain. Das Löschen der CORS-Richtlinie für einen Container entfernt Berechtigungen für ursprungsübergreifende Anfragen.

Löschen einer CORS-Richtlinie (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) den Namen des Containers, für den Sie eine CORS-Richtlinie löschen möchten.

   Die Seite mit den Containerdetails wird angezeigt.
3. Wählen Sie im Abschnitt Container CORS policy (Container-CORS-Richtlinie) die Option Delete CORS policy (CORS-Richtlinie löschen).
4. Wählen Sie zur Bestätigung Weiter und anschließend Speichern) aus.

Löschen einer CORS-Richtlinie (AWS CLI)

- Verwenden Sie in der AWS CLI den Befehl `delete-cors-policy`:

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

  Dieser Befehl hat keinen Rückgabewert.

Fehlerbehebung bei CORS-Problemen

Wenn Sie beim Zugriff auf einen Container mit einer CORS-Richtlinie auf unerwartetes Verhalten stoßen, gehen Sie wie folgt vor, um das Problem zu beheben.

1. Vergewissern Sie sich, dass die CORS-Richtlinie dem Container zugewiesen ist.

   Detaillierte Anweisungen finden Sie unter the section called “Anzeigen einer CORS-Richtlinie”.
2. Erfassen Sie die vollständige Anfrage und die Antwort mit einem Tool Ihrer Wahl (z. B. der Entwicklerkonsole Ihres Browsers). Vergewissern Sie sich, dass die CORS-Richtlinie, die dem

Container zugewiesen ist, mindestens eine CORS-Regel enthält, die mit Daten in Ihrer Anfrage übereinstimmt, wie folgt:

a. Stellen Sie sicher, dass die Anfrage einen `Origin`-Header besitzt.

   Wenn kein -Header vorhanden ist,MediaStoreverarbeitet die Anfrage nicht als ursprungsübergreifende Anfrage und sendet in der Antwort keine CORS-Antwort-Header zurück.

b. Stellen Sie sicher, dass der `Origin`-Header in Ihrer Anfrage mit mindestens einem der `AllowedOrigins`-Elemente in der betreffenden `CORSRule` übereinstimmt.

   Das Schema, der Host und die Port-Werte im `Origin`-Anfrageheader müssen mit den `AllowedOrigins` in der `CORSRule` übereinstimmen. Wenn Sie beispielsweise die `CORSRule` so eingerichtet haben, dass der Ursprung `http://www.example.com` zulässig ist, stimmen die Ursprünge `https://www.example.com` und `http://www.example.com:80` in Ihrer Anfrage nicht mit dem in Ihrer Konfiguration erlaubten Ursprung überein.

c. Stellen Sie sicher, dass die Methode in Ihrer Anfrage (oder die in `Access-Control-Request-Method` spezifizierte Methode, falls es sich um eine Preflight-Anfrage handelt) eines der `AllowedMethods`-Elemente in derselben `CORSRule` ist.

d. Wenn bei einer Preflight-Anfrage die anfrage einen `Access-Control-Request-Headers`-Header enthält, überprüfen Sie, ob die `CORSRule` die `AllowedHeaders`-Einträge für jeden Wert im `Access-Control-Request-Headers`-Header enthält.

CORS-Beispielrichtlinien

Die folgenden Beispiele zeigen Cross-Origin Resource Sharing (CORS)-Richtlinien.

Themen

CORS-Beispielrichtlinien: Lesezugriff für jede Domäne

Die folgende Richtlinie erlaubt es einer Webseite aus einer beliebigen Domäne, Inhalte aus Ihrem AWS Elemental abzurufenMediaStore-Container. Die Anfrage enthält alle HTTP-Header der

Ursprungsdomäne, und der Service reagiert nur auf HTTP-GET- und HTTP-HEAD-Anfragen aus der Ursprungsdomäne. Die Ergebnisse werden für 3.000 Sekunden zwischengespeichert, bevor eine neue Ergebnismenge bereitgestellt wird.

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

CORS-Beispielrichtlinien: Lesezugriff für eine bestimmte Domäne

Die folgende Richtlinie erlaubt es einer Webseite aus`https://www.example.com`um Inhalte von Ihrem AWS Elementar abzurufenMediaStore-Container. Die Anfrage enthält alle HTTP-Header aus `https://www.example.com`, und der Service reagiert nur auf HTTP-GET- und HTTP-HEAD-Anfragen aus `https://www.example.com`. Die Ergebnisse werden für 3.000 Sekunden zwischengespeichert, bevor eine neue Ergebnismenge bereitgestellt wird.

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

```
]
```

Objektlebenszyklus-Richtlinien in AWS ElementarMediaStore

Sie können für jeden Container eine Objektlebenszyklus-Richtlinie erstellen, die regelt, wie lange Objekte im Container gespeichert werden sollen. Wenn Objekte das angegebene Höchstalter erreichen, wird AWS ElementarMediaStorelöscht die Objekte. Sie können Objekte löschen, nachdem sie nicht mehr benötigt werden, um Speicherkosten sparen.

Sie können auch festlegen, dass MediaStore Objekte nach Erreichen eines bestimmten Alters in die Speicherklasse für den seltenen Zugriff verschieben soll. Objekte, die in der Speicherklasse für seltenen Zugriff gespeichert sind, weisen andere Speicher- und Abrufraten auf als Objekte, die in der Standardspeicherklasse gespeichert sind. Weitere Informationen finden Sie unter MediaStore- Preise.

Eine Objektlebenszyklus-Richtlinie enthält Regeln, die die Lebensdauer von Objekten anhand von Unterordnern bestimmen. (Sie können eine Objektlebenszyklus-Richtlinie keinen einzelnen Objekten zuweisen). Sie können nur eine Objektlebenszyklus-Richtlinie an einen Container anhängen, aber Sie können bis zu 10 Regeln zu jeder Objektlebenszyklus-Richtlinie hinzufügen. Weitere Informationen finden Sie unter Komponenten einer Objektlebenszyklus-Richtlinie .

Themen

Komponenten einer Objektlebenszyklus-Richtlinie

Objektlebenszyklus-Richtlinien steuern, wie lange Objekte in einem AWS Elementar bleibenMediaStore-Container. Jede Objektlebenszyklus-Richtlinie besteht aus mindestens einer Regel, die die Lebensdauer von Objekten bestimmt. Eine Regel kann für einen Ordner, mehrere Ordner oder den gesamten Container gelten.

Sie können eine Objektlebenszyklus-Richtlinie an einen Container anhängen und jede Objektlebenszyklus-Richtlinie kann bis zu 10 Regeln enthalten. Sie können eine Objektlebenszyklus-Richtlinie keinem einzelnen Objekt zuweisen.

Regeln in einer Objektlebenszyklus-Richtlinie

Sie können drei Arten von Regeln erstellen:

- Transiente Daten
- Objekt löschen
- Lebenszyklusübergang

Transiente Daten

Eine Regel für transiente Daten legt fest, dass Objekte innerhalb von Sekunden ablaufen. Dieser Regeltyp gilt nur für Objekte, die dem Container hinzugefügt werden, nachdem die Richtlinie wirksam wird. Es dauert bis zu 20 Minuten, bis MediaStore die neue Richtlinie auf den Container anwendet.

Hier sehen Sie ein Beispiel für eine Regel für transiente Daten:

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

Regeln für transiente Daten bestehen aus drei Teilen:

- `path`: Immer auf `wildcard` gesetzt. Mit diesem Teil definieren Sie, welche Objekte gelöscht werden sollen. Sie können einen oder mehrere Platzhalter verwenden, dargestellt durch ein Sternchen (*). Jeder Platzhalter steht für eine beliebige Kombination aus null oder mehr Zeichen. Beispielsweise gilt `"path": [ {"wildcard": "Football/index*.m3u8"} ],` für alle Dateien im Ordner `Football`, die dem Muster von `index*.m3u8` entsprechen (z. B. „index.m3u8“, „index1.m3us8“ und „index123456.m3u8“). Sie können bis zu 10 -Pfade in eine Regel aufnehmen.

- `seconds_since_create`: Immer auf `numeric` gesetzt. Sie können einen Wert von 1 bis 300 angeben. Sie können den Operator auch auf „größer als“ (>) oder „größer oder gleich“ (>=) festlegen.
- `action`: Immer auf `EXPIRE` gesetzt.

Bei Regeln für transiente Daten (Objekte laufen innerhalb von Sekunden ab) gibt es keine Verzögerung zwischen dem Ablauf eines Objekts und dem Löschen des Objekts.

> **Note**
>
> Objekte, die einer Regel für transiente Daten unterliegen, sind nicht in einer `list-items`-Antwort enthalten. Darüber hinaus emittieren Objekte, die aufgrund einer transienten Datenregel ablaufen, keineCloudWatch-Ereignis, wenn sie ablaufen.

Objekt löschen

Eine Regel zum Löschen von Objekten legt fest, dass Objekte innerhalb von Tagen ablaufen. Dieser Regeltyp gilt für alle Objekte im Container, auch wenn sie dem Container hinzugefügt wurden, bevor die Richtlinie erstellt wurde. Es dauert bis zu 20 Minuten, bis MediaStore die neue Richtlinie anwendet, aber es kann bis zu 24 Stunden dauern, bis die Objekte aus dem Container gelöscht werden.

Ein Beispiel für zwei Regeln zum Löschen von Objekten sieht wie folgt aus:

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

```
            ]
        },
        "action": "EXPIRE"
    }
```

Regeln zum Löschen von Objekten bestehen aus drei Teilen:

- `path`: Festlegung entweder auf `prefix` oder `wildcard`. `prefix` und `wildcard` können in derselben Regel nicht zusammen verwendet werden. Wenn Sie beide verwenden möchten, müssen Sie, wie im obigen Beispiel gezeigt, eine Regel für `prefix` und eine separate Regel für `wildcard`erstellen.
  - `prefix` - Legen Sie den Pfad auf `prefix` fest, wenn Sie alle Objekte innerhalb eines bestimmten Ordners löschen möchten. Wenn der Parameter leer ist (`"path": [ { "prefix": "" } ],`), umfasst das Ziel alle Objekte, die an einer beliebigen Stelle innerhalb des aktuellen Containers gespeichert sind. Sie können bis zu 10 `prefix`-Pfade in eine Regel aufnehmen.
  - `wildcard` - Legen Sie den Pfad auf `wildcard` fest, wenn Sie bestimmte Objekte basierend auf Dateinamen und/oder Dateityp löschen möchten. Sie können einen oder mehrere Platzhalter verwenden, dargestellt durch ein Sternchen (*). Jeder Platzhalter steht für eine beliebige Kombination aus null oder mehr Zeichen. `"path": [ {"wildcard": "Football/*.ts"} ],` gilt z. B. für alle Dateien im `Football`-Ordner, die dem Muster von `*.ts` entsprechen (z. B. Dateiname.ts, Dateiname1.ts und Dateiname123456.ts). Sie können bis zu 10 `wildcard`-Pfade in eine Regel aufnehmen.
- `days_since_create`: Immer auf `numeric` gesetzt. Sie können einen Wert von 1 bis 36.500 Tage angeben. Sie können den Operator auch auf „größer als“ (**>**) oder „größer oder gleich“ (**>=**) festlegen.
- `action`: Immer auf `EXPIRE` gesetzt.

Bei Regeln zum Löschen von Objekten (Objekte laufen innerhalb von Tagen ab) gibt es u. U. eine geringfügige Verzögerung zwischen dem Ablauf eines Objekts und dem Löschen des Objekts. Änderungen bei der Fakturierung erfolgen jedoch, sobald das Objekt abläuft. Wenn beispielsweise eine Lebenszyklusregel 10 angibt `days_since_create`, wird dem Konto das Objekt nicht in Rechnung gestellt, nachdem das Objekt 10 Tage alt ist, auch wenn das Objekt noch nicht gelöscht wurde.

Lebenszyklusübergang

Eine Lebenszyklus-Übergangsregel legt fest, dass Objekte in die Speicherklasse für den seltenen Zugriff verschoben werden, nachdem sie ein bestimmtes Alter (gemessen in Tagen) erreicht haben. Objekte, die in der Speicherklasse für seltenen Zugriff gespeichert sind, weisen andere Speicher- und Abrufraten auf als Objekte, die in der Standardspeicherklasse gespeichert sind. Weitere Informationen finden Sie unter MediaStore- Preise.

Sobald ein Objekt in die Speicherklasse für seltenen Zugriff verschoben wurde, können Sie es nicht zurück in die Standardspeicherklasse verschieben.

Die Lebenszyklusübergangsregel gilt für alle Objekte im Container, selbst wenn sie dem Container hinzugefügt wurden, bevor die Richtlinie erstellt wurde. Es dauert bis zu 20 Minuten, bis MediaStore die neue Richtlinie anwendet, aber es kann bis zu 24 Stunden dauern, bis die Objekte aus dem Container gelöscht werden.

Ein Beispiel für eine Lebenszyklus-Übergangsregel sieht folgendermaßen aus:

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

Lebenszyklus-Übergangsregeln haben drei Teile:

- `path`: Festlegung entweder auf `prefix` oder `wildcard`. `prefix` und `wildcard` können in derselben Regel nicht zusammen verwendet werden. Wenn Sie beide verwenden möchten, müssen Sie eine Regel für `prefix` und eine separate Regel für `wildcard` erstellen.
  - `prefix`: Sie legen den Pfad auf `prefix` fest, wenn Sie alle Objekte in einem bestimmten Ordner in die Speicherklasse für seltenen Zugriff übertragen möchten. Wenn der Parameter leer ist (`"path": [ { "prefix": "" } ],`), umfasst das Ziel alle Objekte, die an einer beliebigen Stelle innerhalb des aktuellen Containers gespeichert sind. Sie können bis zu 10 `prefix`-Pfade in eine Regel aufnehmen.

- `wildcard`: Sie legen den Pfad auf `wildcard` fest, wenn Sie bestimmte Objekte basierend auf dem Dateinamen und/oder Dateityp in die Speicherklasse für seltenen Zugriff übertragen möchten. Sie können einen oder mehrere Platzhalter verwenden, dargestellt durch ein Sternchen (*). Jeder Platzhalter steht für eine beliebige Kombination aus null oder mehr Zeichen. `"path": [ {"wildcard": "Football/*.ts"} ],` gilt z. B. für alle Dateien im `Football`-Ordner, die dem Muster von `*.ts` entsprechen (z. B. Dateiname.ts, Dateiname1.ts und Dateiname123456.ts). Sie können bis zu 10 `wildcard`-Pfade in eine Regel aufnehmen.

- `days_since_create`: Immer auf `"numeric": [">=" , 30]` gesetzt.
- `action`: Immer auf `ARCHIVE` gesetzt.

Beispiel

Angenommen, ein Container mit dem Namen `LiveEvents` verfügt über vier Unterordner: `Football`, `Baseball`, `Basketball` und `AwardsShow`. Die dem `LiveEvents`-Ordner zugewiesene Objektlebenszyklus-Richtlinie kann wie folgt aussehen:

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

Die obige Richtlinie legt Folgendes fest:

- Die erste Regel weist AWS Elemental anMediaStoreum Objekte zu löschen, die in der`LiveEvents/Football`folder und die`LiveEvents/Baseball`Ordner, nachdem sie älter als 28 Tage sind.
- Die zweite Regel weist den Service an, Objekte zu löschen, die im Ordner `LiveEvents/AwardsShow` gespeichert sind, nachdem sie mindestens 15 Tage alt sind.
- Die dritte Regel weist den Service an, Objekte zu löschen, die an einem beliebigen Speicherort im Container `LiveEvents` gespeichert sind, nachdem sie älter als 40 Tage sind. Diese Regel gilt für Objekte, die direkt im `LiveEvents`-Container gespeichert sind, sowie für gespeicherte Objekte in jedem der vier Unterordner des Containers.
- Die vierte Regel weist den Service an, Objekte im `Football`-Ordner zu löschen, die dem Muster `*.ts`-entsprechen, nachdem sie älter als 20 Tage sind.
- Die fünfte Regel weist den Dienst an, Objekte im`Football`folder, der dem Muster entspricht`index*.m3u8`nachdem sie älter als 15 Sekunden sind. MediaStorelöscht diese Dateien 16 Sekunden nachdem sie in den Container abgelegt wurden.
- Die sechste Regel weist den Service an, Objekte im `Program`-Ordner in die Speicherklasse für seltenen Zugriff zu verschieben, nachdem sie 30 Tage alt sind.

Weitere Beispiele für Objektlebenszyklus-Richtlinien finden Sie unter Beispiele für Objektlebenszyklus-Richtlinien.

Hinzufügen einer Objektlebenszyklus-Richtlinie zu einem Container

Mit einer Objektlebenszyklus-Richtlinie können Sie angeben, wie lange Ihre Objekte in einem Container gespeichert werden sollen. Sie legen ein Ablaufdatum fest. Nach AWS DatumMediaStorelöscht die Objekte. Es dauert bis zu 20 Minuten, bis der Service die neue Richtlinie auf den Container anwendet.

Weitere Informationen zum Erstellen einer Lebenszyklusrichtlinie finden Sie unter Komponenten einer Objektlebenszyklus-Richtlinie.

> **Note**
>
> Bei Regeln zum Löschen von Objekten (Objekte laufen innerhalb von Tagen ab) gibt es u. U. eine geringfügige Verzögerung zwischen dem Ablauf eines Objekts und dem Löschen des Objekts. Änderungen bei der Fakturierung erfolgen jedoch, sobald das Objekt abläuft. Wenn beispielsweise eine Lebenszyklusregel 10 angibt `days_since_create`, wird dem Konto das

Objekt nicht in Rechnung gestellt, nachdem das Objekt 10 Tage alt ist, auch wenn das Objekt noch nicht gelöscht wurde.

So fügen Sie eine Objektlebenszyklus-Richtlinie hinzu (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) den Namen des Containers, für den Sie eine Objektlebenszyklus-Richtlinie erstellen möchten.

   Die Seite mit den Containerdetails wird angezeigt.
3. Wählen Sie im Abschnitt Objektlebenszyklus-Richtlinie die Option zum Erstellen einer Objektlebenszyklus-Richtlinie aus.
4. Fügen Sie die Richtlinie im JSON-Format ein und wählen Sie Save (Speichern).

So fügen Sie eine Objektlebenszyklus-Richtlinie hinzu (AWS CLI)

1. Erstellen Sie eine Datei, die die Objektlebenszyklus-Richtlinie definiert:

```
{
    "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. Verwenden Sie in der AWS CLI den Befehl `put-lifecycle-policy`:

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

Dieser Befehl hat keinen Rückgabewert. Der Service fügt die angegebene Richtlinie an den Container an.

Anzeigen einer Objektlebenszyklus-Richtlinie

Eine Objektlebenszyklus-Richtlinie gibt an, wie lange Objekte in einem Container aufbewahrt werden sollen.

So zeigen Sie eine Objektlebenszyklus-Richtlinie an (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) den Namen des Containers, für den Sie die Objektlebenszyklus-Richtlinie anzeigen möchten.

   Die Container-Detailseite wird geöffnet und die Objektlebenszyklus-Richtlinie wird im Abschnitt Objektlebenszyklus-Richtlinie angezeigt.

So zeigen Sie eine Objektlebenszyklus-Richtlinie an (AWS CLI)

- Verwenden Sie in der AWS CLI den Befehl `get-lifecycle-policy`:

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

Im folgenden Beispiel finden Sie den Rückgabewert:

```
{
    "LifecyclePolicy": "{
```

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

Bearbeiten einer Objektlebenszyklus-Richtlinie

Sie können keine vorhandene Objektlebenszyklus-Richtlinie bearbeiten. Sie können jedoch eine vorhandene Richtlinie ändern, indem Sie eine Ersatz-Richtlinie hochladen. Es dauert bis zu 20 Minuten, bis der Service die aktualisierte Richtlinie auf den Container anwendet.

So bearbeiten Sie eine Objektlebenszyklus-Richtlinie (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) den Namen des Containers, für den Sie die Objektlebenszyklus-Richtlinie bearbeiten möchten.

   Die Seite mit den Containerdetails wird angezeigt.
3. Wählen Sie im Abschnitt Objektlebenszyklus-Richtlinie die Option zum Bearbeiten einer Objektlebenszyklus-Richtlinie aus.
4. Nehmen Sie Ihre Änderungen an der Richtlinie vor und wählen Sie Save (Speichern).

So bearbeiten Sie eine Objektlebenszyklus-Richtlinie (AWS CLI)

1. Erstellen Sie eine Datei, die die aktualisierte Objektlebenszyklus-Richtlinie definiert:

```
{
```

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

2. Verwenden Sie in der AWS CLI den Befehl `put-lifecycle-policy`:

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

   Dieser Befehl hat keinen Rückgabewert. Der Service fügt die angegebene Richtlinie an den Container an und ersetzt damit die vorherige Richtlinie.

Löschen einer Objektlebenszyklus-Richtlinie

Wenn Sie eine Objekt-Lebenszyklus-Richtlinie löschen, dauert es bis zu 20 Minuten, bis der Service die Änderung auf den Container angewendet hat.

So löschen Sie eine Objektlebenszyklus-Richtlinie (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) den Namen des Containers, für den Sie die Objektlebenszyklus-Richtlinie löschen möchten.

   Die Seite mit den Containerdetails wird angezeigt.
3. Wählen Sie im Abschnitt Objektlebenszyklus-Richtlinie die Option zum Löschen einer Lebenszyklusrichtlinie aus.
4. Wählen Sie zur Bestätigung Weiter und anschließend Speichern) aus.

So löschen Sie eine Objektlebenszyklus-Richtlinie (AWS CLI)

- Verwenden Sie in der AWS CLI den Befehl `delete-lifecycle-policy`:

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

Dieser Befehl hat keinen Rückgabewert.

Beispiele für Objektlebenszyklus-Richtlinien

Die folgenden Beispiele zeigen Objekt-Lebenszyklusrichtlinien.

Themen

Beispiel für Objektlebenszyklus-Richtlinie: Ablauf innerhalb von Sekunden

Die folgende Richtlinie legt fest, dass MediaStore Objekte löscht, die alle folgenden Kriterien erfüllen:

- Das Objekt wird dem Container hinzugefügt, nachdem die Richtlinie wirksam wurde.
- Das Objekt wird im `Football`-Ordner gespeichert.
- Das Objekt hat die Dateierweiterung `m3u8`.
- Das Objekt befindet sich seit mehr als 20 Sekunden im Container.

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

```
                "seconds_since_create": [
                    {"numeric": [ ">", 20 ]}
                ]
            },
            "action": "EXPIRE"
        }
    ]
}
```

Beispiel für Objektlebenszyklus-Richtlinie: Ablauf innerhalb von Tagen

Die folgende Richtlinie legt fest, dass MediaStore Objekte löscht, die alle folgenden Kriterien erfüllen:

- Das Objekt wird im `Program`-Ordner gespeichert.
- Das Objekt hat die Dateierweiterung `ts`.
- Das Objekt befindet sich seit mehr als 5 Tagen im Container.

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

Beispiel für Objektlebenszyklus-Richtlinie: Umstellung auf Speicherklasse mit seltenem Zugriff

Die folgende Richtlinie legt fest, dass MediaStore Objekte in die Speicherklasse für seltenen Zugriff verschiebt, wenn sie 30 Tage alt sind. Objekte, die in der Speicherklasse für seltenen Zugriff gespeichert sind, weisen andere Speicher- und Abrufraten auf als Objekte, die in der Standardspeicherklasse gespeichert sind.

Das `days_since_create`-Feld muss auf `"numeric": [">=" ,30]` eingestellt sein.

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

Beispiel für Objektlebenszyklus-Richtlinie: Mehrere Regeln

Die folgende Richtlinie legt fest, dass MediaStore Folgendes ausführt:

- Objekte, die im `AwardsShow`-Ordner gespeichert sind, nach 30 Tagen in die Speicherklasse für den seltenen Zugriff verschieben
- Objekte mit der Dateierweiterung `m3u8` nach 20 Sekunden im `Football`-Ordner löschen
- Objekte nach 10 Tagen im `April`-Ordner löschen
- Objekte mit der Dateierweiterung `ts` nach 5 Tagen im `Program`-Ordner löschen

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

Beispiel für Objektlebenszyklus-Richtlinie: Container leeren

Die folgende Objektlebenszyklus-Richtlinie legt fest, dass MediaStore alle Objekte im Container, einschließlich Ordner und Unterordner, 1 Tag nach dem Hinzufügen zum Container löscht. Wenn der Container Objekte enthält, bevor diese Richtlinie angewendet wird,MediaStorelöscht die Objekte

1 Tag nach dem Inkrafttreten der Richtlinie. Es dauert bis zu 20 Minuten, bis der Service die neue Richtlinie auf den Container anwendet.

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

Metrische Richtlinien in AWS Elemental MediaStore

Für jeden Container können Sie eine Metrikrichtlinie hinzufügen, damit AWS Elemental Metriken MediaStore an Amazon senden kann CloudWatch. Es dauert bis zu 20 Minuten, bis die neue Richtlinie wirksam wird. Eine Beschreibung der einzelnen MediaStore Metriken finden Sie unterMediaStore Metriken.

Eine Metrikrichtlinie enthält Folgendes:

- Eine Einstellung zum Aktivieren oder Deaktivieren von Metriken auf Containerebene.
- Zwischen null und fünf Regeln, die Metriken auf Objektebene aktivieren. Wenn die Richtlinie Regeln enthält, muss jede Regel Folgendes umfassen:
  - Eine Objektgruppe, die definiert, welche Objekte in die Gruppe aufgenommen werden sollen. Die Definition kann ein Pfad oder ein Dateiname sein, darf jedoch nicht mehr als 900 Zeichen enthalten. Gültige Zeichen sind: a–z, A–Z, 0–9, _ (Unterstrich), = (gleich), : (Doppelpunkt), . (Punkt), - (Bindestrich), ~ (Tilde), / (Schrägstrich) und * (Sternchen). Platzhalter (*) sind zulässig.
  - Ein Objektgruppenname, mit dem Sie auf die Objektgruppe verweisen können. Der Name darf nicht mehr als 30 Zeichen enthalten. Gültige Zeichen sind a–z, A–Z, 0–9 und _ (Unterstrich).

Wenn ein Objekt mehreren Regeln entspricht, CloudWatch wird für jede passende Regel ein Datenpunkt angezeigt. Wenn ein Objekt beispielsweise zwei Regeln mit dem Namen`rule1` und entspricht`rule2`, CloudWatch werden zwei Datenpunkte für diese Regeln angezeigt. Die erste hat die Dimension `ObjectGroupName=rule1` und die zweite die Dimension `ObjectGroupName=rule2`.

Themen

Hinzufügen einer Metrikrichtlinie

Eine Metrikrichtlinie enthält Regeln, die festlegen, welche Metriken AWS Elemental MediaStore an Amazon sendet CloudWatch. Beispiele für Metrikrichtlinien finden Sie unter Beispiele für Metrikrichtlinien.

So fügen Sie eine Metrikrichtlinie hinzu (Konsole)

1. Öffnen Sie die MediaStore Konsole unter https://console.aws.amazon.com/mediastore/.
2. Wählen Sie auf der Seite Containers (Container) den Namen des Containers, dem Sie eine Metrikrichtlinie hinzufügen möchten.

   Die Seite mit den Containerdetails wird angezeigt.
3. Wählen Sie im Abschnitt Metric policy (Metrikrichtlinie) die Option Create metric policy (Metrikrichtlinie erstellen) aus.
4. Fügen Sie die Richtlinie im JSON-Format ein und wählen Sie Save (Speichern).

Anzeigen einer Metrikrichtlinie

Sie können die Konsole oder die AWS CLI verwenden, um die Metrikrichtlinie eines Containers anzuzeigen.

So zeigen Sie eine Metrikrichtlinie an (Konsole)

1. Öffnen Sie die MediaStore Konsole unter https://console.aws.amazon.com/mediastore/.

2. Wählen Sie auf der Seite Containers (Container) den Containernamen aus.

   Die Seite mit den Containerdetails wird angezeigt. Die Richtlinie wird im Abschnitt Metric policy (Metrikrichtlinie) angezeigt.

Bearbeiten einer Metrikrichtlinie

Eine Metrikrichtlinie enthält Regeln, die festlegen, welche Metriken AWS Elemental MediaStore an Amazon sendet CloudWatch. Wenn Sie eine vorhandene Metrikrichtlinie bearbeiten, dauert es bis zu 20 Minuten, bis die neue Richtlinie wirksam wird. Beispiele für Metrikrichtlinien finden Sie unter Beispiele für Metrikrichtlinien.

So bearbeiten Sie eine Metrikrichtlinie (Konsole)

1. Öffnen Sie die MediaStore Konsole unter https://console.aws.amazon.com/mediastore/.
2. Wählen Sie auf der Seite Containers (Container) den Containernamen aus.
3. Wählen Sie im Abschnitt Metric policy (Metrikrichtlinie) die Option Edit metric policy (Metrikrichtlinie bearbeiten) aus.
4. Nehmen Sie die entsprechenden Änderungen vor und wählen Sie dann Save (Speichern) aus.

Beispiele für Metrikrichtlinien

Die folgenden Beispiele veranschaulichen Metrikrichtlinien, die für verschiedene Anwendungsfälle erstellt wurden.

Themen

- Beispiel für Metrikrichtlinien: Metriken auf Containerebene
- Beispiel für Metrikrichtlinien: Metriken auf Pfadebene
- Beispiel für Metrikrichtlinien: Metriken auf Container- und Pfadebene
- Beispiel für Metrikrichtlinien: Metriken auf Pfadebene mit Platzhaltern
- Beispiel für Metrikrichtlinien: Metriken auf Pfadebene mit sich überschneidenden Regeln

Beispiel für Metrikrichtlinien: Metriken auf Containerebene

Diese Beispielrichtlinie besagt, dass AWS Elemental Metriken auf Container-Ebene CloudWatch an Amazon senden MediaStore sollte. Dies schließt beispielsweise die Metrik `RequestCount` ein, die

die Anzahl der `Put`-Anforderungen an den Container zählt. Alternativ können Sie diese Richtlinie auf `DISABLED` einstellen.

Da diese Richtlinie keine Regeln MediaStore enthält, werden keine Metriken auf Pfadebene gesendet. Beispielsweise können Sie nicht sehen, wie viele `Put`-Anforderungen an einen bestimmten Ordner innerhalb dieses Containers gestellt wurden.

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

Beispiel für Metrikrichtlinien: Metriken auf Pfadebene

Diese Beispielrichtlinie besagt, dass AWS Elemental keine Metriken auf Container-Ebene CloudWatch an Amazon senden MediaStore sollte. Darüber hinaus soll MediaStore Metriken für Objekte in zwei bestimmten Ordnern senden: `baseball/saturday` und `football/saturday`. Die Metriken für MediaStore -Anforderungen lauten wie folgt:

- Anfragen an den`baseball/saturday` Ordner haben eine CloudWatch Dimension von`ObjectGroupName=baseballGroup`.
- Anforderungen an den Ordner `football/saturday` haben die Dimension `ObjectGroupName=footballGroup`.

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

Beispiel für Metrikrichtlinien: Metriken auf Container- und Pfadebene

Diese Beispielrichtlinie besagt, dass AWS Elemental Metriken auf Container-Ebene CloudWatch an Amazon senden MediaStore sollte. Außerdem MediaStore sollten Metriken für Objekte in zwei bestimmten Ordnern gesendet werden:`baseball/saturday` und`football/saturday`. Die Metriken für MediaStore-Anforderungen lauten wie folgt:

- Anfragen an den`baseball/saturday` Ordner haben eine CloudWatch Dimension von`ObjectGroupName=baseballGroup`.
- Anfragen an den`football/saturday` Ordner haben eine CloudWatch Dimension`ObjectGroupName=footballGroup`.

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

Beispiel für Metrikrichtlinien: Metriken auf Pfadebene mit Platzhaltern

Diese Beispielrichtlinie besagt, dass AWS Elemental Metriken auf Container-Ebene CloudWatch an Amazon senden MediaStore sollte. Darüber hinaus MediaStore sollte es auch Metriken für Objekte senden, die auf ihrem Dateinamen basieren. Ein Platzhalter gibt an, dass die Objekte an einer beliebigen Stelle im Container gespeichert werden und einen beliebigen Dateinamen haben können, solange dieser mit der Erweiterung `.m3u8` endet.

```
{
   "ContainerLevelMetrics": "ENABLED",
   "MetricPolicyRules": [
     {
       "ObjectGroup": "*.m3u8",
```

```
      "ObjectGroupName": "index"
    }
  ]
}
```

Beispiel für Metrikrichtlinien: Metriken auf Pfadebene mit sich überschneidenden Regeln

Diese Beispielrichtlinie besagt, dass AWS Elemental Metriken auf Container-Ebene CloudWatch an Amazon senden MediaStore sollte. Außerdem MediaStore sollten Metriken für zwei Ordner gesendet werden:`sports/football/saturday` und`sports/football`.

Die Metriken für MediaStore Anfragen an den`sports/football/saturday` Ordner haben eine CloudWatch Dimension von`ObjectGroupName=footballGroup1`. Da Objekte, die im Ordner `sports/football` gespeichert sind, beiden Regeln entsprechen, zeigt CloudWatch zwei Datenpunkte für diese Objekte an: einen mit der Dimension `ObjectGroupName=footballGroup1` und den zweiten mit der Dimension `ObjectGroupName=footballGroup2`.

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

Ordner in AWS ElementalMediaStore

Ordner sind Teilbereiche in einem Container. Sie Verwenden Ordner, um Ihren Container so zu unterteilen, wie Sie Unterordner erstellen, um einen Ordner in einem Dateisystem zu unterteilen. Sie können bis zu 10 Ordnerebenen erstellen (ohne den Container selbst).

Ordner sind optional. Sie können auswählen, ob Sie die Objekte direkt in einen Container statt in einen Ordner hochladen wollen. Ordner bieten jedoch eine einfache Möglichkeit, Ihre Objekte zu organisieren.

Um ein Objekt in einen Ordner hochzuladen, geben Sie den Pfad zum Ordner an. Wenn der Ordner bereits vorhanden ist, AWS ElementalMediaStorespeichert das Objekt im Ordner. Wenn der Ordner nicht vorhanden, legt der Service ihn an und speichert das Objekt in dem Ordner.

Nehmen wir beispielsweise an, dass Sie einen Container mit dem Namen haben`movies`, und Sie laden eine Datei mit dem Namen hoch`mlaw.ts`mit dem Pfad`premium/canada`aus. AWS ElementalMediaStoreSpeichert das Objekt im Unterordner canada unter dem Ordner premium. Wenn keine der Ordner vorhanden ist, erstellt der Service sowohl den Ordner `premium`, als auch den Unterordner `canada`, und speichert das Objekt im Unterordner `canada`. Wenn Sie nur den Container `movies` (ohne Pfad) angeben, speichert der Service das Objekt direkt im Container.

AWS ElementalMediaStorelöscht automatisch einen Ordner, wenn Sie das letzte Objekt in diesem Ordner löschen. Der Service löscht auch alle leeren Ordner oberhalb dieses Ordners. Beispielsweise angenommen, Sie verfügen über einen Ordner mit dem Namen premium, der keine Dateien enthält, jedoch einen Unterordner mit dem Namen `canada`. Der Unterordner `canada` enthält eine mit dem Namen .`mlaw.ts` Wenn Sie die Datei `mlaw.ts` löschen, löscht der Service die Ordner `premium` und `canada`. Dieses automatische Löschen gilt nur für Ordner. Der Service löscht keine leeren Container.

Themen

Regeln für Ordnernamen

Wenn Sie einen Namen für Ihren Ordner wählen, beachten Sie Folgendes:

- Der Name darf nur folgende Zeichen enthalten: Großbuchstaben (A-Z), Kleinbuchstaben (a-z), Ziffern (0-9), Punkte (.), Bindestriche (_), Bindestriche (_), Punkte (.), Bindestriche (_), Gleichheitszeichen (=) und Doppelpunkte (:).
- Der Name muss mindestens ein Zeichen lang sein. Leere Ordnernamen (wie`folder1//folder3/`) sind nicht erlaubt.
- Bei den Namen muss die Groß- und Kleinschreibung beachtet werden. Sie können beispielsweise einen Ordner mit dem Namen `myFolder` und einen Ordner mit dem Namen `myfolder` im selben Container oder Ordner verwenden, weil diese Namen eindeutig sind.
- Der Name muss eindeutig innerhalb seines übergeordneten Containers oder Ordners sein. Sie können beispielsweise einen Ordner mit dem Namen `myfolder` in zwei verschiedenen Containern erstellen: `movies/myfolder` und `sports/myfolder`.
- Der Name kann denselben Namen wie der übergeordnete Container haben.
- Der Ordner kann nach dem Erstellen nicht mehr umbenannt werden.

Erstellen eines Ordners

Sie können Ordner erstellen, wenn Sie Objekte hochladen. Um ein Objekt in einen Ordner hochzuladen, geben Sie den Pfad zum Ordner an. Wenn der Ordner bereits vorhanden ist, AWS ElementalMediaStorespeichert das Objekt im Ordner. Wenn der Ordner nicht vorhanden, legt der Service ihn an und speichert das Objekt in dem Ordner.

Weitere Informationen finden Sie unter the section called “Hochladen eines Objekts” .

Löschen eines Ordners

Sie können Ordner nur löschen, wenn der Ordner leer ist. Sie können keinen Ordner löschen, die Objekte enthalten.

AWS ElementalMediaStorelöscht automatisch einen Ordner, wenn Sie das letzte Objekt in diesem Ordner löschen. Der Service löscht auch alle leeren Ordner oberhalb dieses Ordners. Beispielsweise angenommen, Sie verfügen über einen Ordner mit dem Namen `premium`, der keine Dateien enthält, jedoch einen Unterordner mit dem Namen `canada`. Der Unterordner `canada` enthält eine mit dem Namen .`mlaw.ts` Wenn Sie die Datei `mlaw.ts` löschen, löscht der Service die Ordner `premium` und `canada`. Dieses automatische Löschen gilt nur für Ordner. Der Service löscht keine leeren Container.

Weitere Informationen finden Sie unter Löschen eines Objekts.

Objekte in AWS ElementalMediaStore

AWS ElementalMediaStore-Komponenten werden alsObjekteaus. Sie können ein Objekt in einen Container oder in einen Ordner innerhalb des Containers hochladen.

In MediaStore können Sie Objekte hochladen, herunterladen und löschen:

- Hochladen – Ein Objekt in einem Container oder Ordner hinzufügen. Dies ist nicht dasselbe wie das Erstellen eines Objekts. Sie müssen Ihre Objekte lokal erstellen, bevor Sie sie auf MediaStore hochladen können.
- Herunterladen – Ein Objekt von MediaStore an einen anderen Speicherort kopieren. Damit wird das Objekt nicht aus MediaStore entfernt.
- Löschen – Ein Objekt aus MediaStore vollständig entfernen. Sie können Objekte einzeln löschen oder einer Objektlebenszyklus-Richtlinie hinzufügen, um Objekte in einem Container nach einer bestimmten Zeit automatisch zu löschen.

MediaStore akzeptiert alle Dateitypen.

Themen

Hochladen eines Objekts

Sie können Objekte in einen Container oder in einen Ordner innerhalb eines Containers hochladen. Um ein Objekt in einen Ordner hochzuladen, geben Sie den Pfad zum Ordner an. Wenn der Ordner bereits vorhanden ist, AWS ElementalMediaStorespeichert das Objekt im Ordner. Wenn der Ordner nicht vorhanden, legt der Service ihn an und speichert das Objekt in dem Ordner. Weitere Informationen über Ordner finden Sie unter Ordner in AWS ElementalMediaStore.

Sie können die MediaStore-Konsole oder die AWS CLI verwenden, um Objekte hochzuladen.

MediaStore unterstützt das Aufteilen der Übertragung von Objekten, wodurch die Latenz reduziert wird, indem ein Objekt zum Download zur Verfügung gestellt wird, während es noch hochgeladen wird. Um diese Funktion zu verwenden, stellen Sie für die Upload-Verfügbarkeit des Objekts `streaming` ein. Sie können den Wert dieses Headers festlegen, wenn Sie das Objekt mithilfe der API hochladen. Wenn Sie in Ihrer Anfrage keine Header angeben, weist MediaStore den Standardwert `standard` für die Uploadverfügbarkeit des Objekts zu.

Objekte dürfen eine Größe von 25 MB für Standard-Upload-Verfügbarkeit und von 10 MB für Streaming-Upload-Verfügbarkeit nicht überschreiten.

> **Note**
>
> Objektdateinamen dürfen nur Buchstaben, Ziffern, Punkte (.), Unterstriche (_), Tilden (~), Bindestriche (-), Gleichheitszeichen (=) und Doppelpunkte (:) enthalten.

Hochladen eines Objekts (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) den Containernamen aus. Der Detailbereich für den Container wird angezeigt.
3. Wählen Sie Upload object (Objekt hochladen).
4. Geben Sie für Target path (Zielpfad) einen Pfad für die Ordner ein. Zum Beispiel `premium/canada`. Wenn einer der Ordner in dem von Ihnen angegebenen Pfad noch nicht vorhanden ist, legt der Service ihn automatisch an.
5. Wählen Sie im Bereich Object (Objekt) die Option Browse (Durchsuchen).
6. Navigieren Sie zum entsprechenden Ordner und wählen Sie ein Objekt zum Hochladen aus.
7. Wählen Sie Open (Öffnen) und anschließend Upload (Hochladen).

> **Note**
>
> Wenn im ausgewählten Ordner bereits eine Datei mit dem gleichen Namen vorhanden ist, ersetzt der Service die Originaldatei durch die hochgeladene Datei.

Ein Objekt hochladen (AWS CLI)

- Verwenden Sie in der AWS CLI den Befehl `put-object`. Sie können auch einen der folgenden Parameter einschließen: `content-type`, `cache-control` (um dem Aufrufer zu erlauben, das Cache-Verhalten des Objekts zu steuern) und `path` (um das Objekt in einen Ordner innerhalb des Containers zu legen).

  > **Note**
  >
  > Nachdem Sie das Objekt hochgeladen haben, können Sie `content-type`, `cache-control` oder `path` nicht mehr bearbeiten.

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

  Im folgenden Beispiel finden Sie den Rückgabewert:

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

Anzeigen einer Liste von Objekten

Sie können das AWS Elementar verwendenMediaStore-Konsole zum Anzeigen von Elementen (Objekten und Ordner), die in der obersten Ebene eines Containers oder in einem Ordner gespeichert sind. Elemente in einem Unterordner des aktuellen Containers oder Ordners werden nicht angezeigt. Mit der AWS CLI können Sie eine Liste von Objekten und Ordnern innerhalb eines Containers anzeigen, unabhängig davon, wie viele Ordner oder Unterordner sich innerhalb des Containers befinden.

Eine Liste der Objekte in einem bestimmten Container anzeigen (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) den Namen des Containers, der den Ordner enthält, den Sie anzeigen möchten.
3. Wählen Sie den Namen des Ordners in der Liste aus.

   Eine Detailseite wird angezeigt. Sie enthält alle Ordner und Objekte, die im Ordner gespeichert sind.

Eine Liste der Objekte in einem bestimmten Ordner anzeigen (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) den Namen des Containers, der den Ordner enthält, den Sie anzeigen möchten.

   Eine Detailseite wird angezeigt. Sie enthält alle Ordner und Objekte, die im Container gespeichert sind.

Eine Liste der Objekte und Ordner in einem bestimmten Container anzeigen (AWS CLI)

- Verwenden Sie in der AWS CLI den Befehl `list-items`:

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

Im folgenden Beispiel finden Sie den Rückgabewert:

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

```
        {
            "Type": "FOLDER",
            "Name": "ExampleLiveDemo"
        }
    ]
}
```

> **Note**
>
> Objekte, die einer `list-items`-Regel unterliegen, sind nicht in einer `seconds_since_create`-Antwort enthalten.

Eine Liste der Objekte und Ordner in einem bestimmten Ordner anzeigen (AWS CLI)

- Verwenden Sie in der AWS CLI den Befehl `list-items` mit dem angegebenen Ordnernamen am Ende der Anfrage:

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

Im folgenden Beispiel finden Sie den Rückgabewert:

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

```
}
```

> **Note**
>
> Objekte, die einer `list-items`-Regel unterliegen, sind nicht in einer `seconds_since_create`-Antwort enthalten.

Anzeigen der Details eines Objekts

Nachdem Sie ein Objekt hochgeladen haben, hat AWS ElementarMediaStoreSpeichert Details wie Änderungsdatum, Inhaltslänge, ETag (Entity-Tag) und Inhaltstyp. Informationen zur Verwendung der Metadaten eines Objekts finden Sie unter Interaktion von MediaStore mit HTTP-Caches.

Anzeigen der Details eines Objekts (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) den Namen des Containers, der das Objekt enthält, das Sie anzeigen möchten.
3. Wenn sich das Objekt, das Sie anzeigen möchten, in einem Ordner befindet, wählen Sie so lange Ordnernamen aus, bis Sie das Objekt sehen.
4. Wählen Sie den Namen des Objekts.

   Eine Detailseite wird angezeigt. Sie enthält Informationen über das Objekt.

Anzeigen der Details eines Objekts (AWS CLI)

- Verwenden Sie in der AWS CLI den Befehl `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
  ```

  Im folgenden Beispiel finden Sie den Rückgabewert:

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

```
    "ContentLength": "2307346",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
}
```

Herunterladen eines Objekts

Sie können die Konsole verwenden, um ein Objekt herunterzuladen. Mit der AWS CLI können Sie ein Objekt oder nur einen Teil eines Objekts herunterladen.

Herunterladen eines Objekts (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) den Namen des Containers, der das Objekt enthält, das Sie herunterladen möchten.
3. Wenn sich das Objekt, das Sie herunterladen möchten, in einem Ordner befindet, wählen Sie so lange Ordnernamen aus, bis Sie das Objekt sehen.
4. Wählen Sie den Namen des Objekts.
5. Wählen Sie auf der Detailseite Object (Objekt) die Option Download (Herunterladen).

Ein Objekt herunterladen (AWS CLI)

- Verwenden Sie in der AWS CLI den Befehl `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
```

Im folgenden Beispiel finden Sie den Rückgabewert:

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

Einen Teil eines Objekts herunterladen (AWS CLI)

- Verwenden Sie in der AWS CLI den Befehl `get-object` und geben Sie einen Bereich an.

```
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
```

Im folgenden Beispiel finden Sie den Rückgabewert:

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

Löschen von Objekten

AWS ElementalMediaStorebietet verschiedene Optionen zum Löschen von Objekten aus Containern:

- Löschen eines einzelnen Objekts. Es fallen keine Gebühren an.
- Leeren eines Containers, um alle Objekte innerhalb eines Containers gleichzeitig zu löschen. Da dieser Prozess API-Aufrufe verwendet, fallen normale API-Gebühren an.
- Hinzufügen einer Objektlebenszyklus-Richtlinie, um Objekte zu löschen, wenn sie ein bestimmtes Alter erreicht haben. Es fallen keine Gebühren an.

Löschen eines Objekts

Sie können die Objekte individuell löschen. Verwenden Sie hierzu die Konsole oder die AWS CLI. Alternativ können Sie eine Objektlebenszyklus-Richtlinie hinzufügen, um Objekte automatisch zu löschen, nachdem sie ein bestimmtes Alter in einem Container erreicht haben. Sie können auch einen Container leeren, um alle Objekte in diesem Container zu löschen.

> **Note**
>
> Wenn Sie das einzige Objekt in einem Ordner löschen, wird AWS ElementalMediaStoreLöscht automatisch den Ordner und alle leeren Ordner oberhalb dieses Ordners. Beispielsweise angenommen, Sie verfügen über einen Ordner mit dem Namen `premium`, der keine Dateien enthält, jedoch einen Unterordner mit dem Namen `canada`. Der Unterordner `canada` enthält eine mit dem Namen `.mlaw.ts` Wenn Sie die Datei `mlaw.ts` löschen, löscht der Service die Ordner `premium` und `canada`.

Löschen eines Objekts (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) den Namen des Containers, der das Objekt enthält, das Sie löschen möchten.
3. Wenn sich das Objekt, das Sie löschen möchten, in einem Ordner befindet, wählen Sie so lange Ordnernamen aus, bis Sie das Objekt sehen.
4. Wählen Sie die Option links neben dem Objektnamen.
5. Wählen Sie Delete (Löschen).

Ein Objekt löschen (AWS CLI)

- Verwenden Sie in der AWS CLI den Befehl `delete-object`.

  Beispiel:

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

  Dieser Befehl hat keinen Rückgabewert.

Leeren eines Containers

Sie können einen Container leeren, um alle Objekte zu löschen, die im Container gespeichert sind. Alternativ können Sie eine Objektlebenszyklus-Richtlinie hinzufügen, um Objekte automatisch zu

löschen, nachdem sie ein bestimmtes Alter in einem Container erreicht haben. Sie können auch Objekte einzeln löschen.

So leeren Sie einen Container (Konsole)

1. Öffnen SieMediaStore-Konsole beihttps://console.aws.amazon.com/mediastore/aus.
2. Wählen Sie auf der Seite Containers (Container) die Option für den Container, den Sie leeren möchten.
3. Wählen Sie Empty container (Container leeren) aus. Es wird eine Bestätigungsmeldung angezeigt.
4. Bestätigen Sie, dass Sie den Container leeren möchten, indem Sie den Containernamen in das Textfeld eingeben, und wählen Sie dannLeeraus.

Sicherheit in AWS Elemental MediaStore

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, auf der AWS Dienste in der ausgeführt AWS Cloud werden. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Weitere Informationen zu den Compliance-Programmen, die für AWS Elemental gelten MediaStore, finden Sie unter AWS Services in Scope by Compliance Program AWS .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Service, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung anwenden können MediaStore. In den folgenden Themen erfahren Sie, wie Sie die Konfiguration vornehmen MediaStore , um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, die Sie bei der Überwachung und Sicherung Ihrer MediaStore Ressourcen unterstützen.

Themen

Datenschutz in AWS Elemental MediaStore

Das AWS Modell der gilt für den Datenschutz in AWS Elemental MediaStore. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der AWS Cloud alle Systeme laufen. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS -Modell der geteilten Verantwortung und in der DSGVO im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie SSL/TLS, um mit Ressourcen zu kommunizieren. AWS Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein. AWS CloudTrail
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-2.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit der Konsole, der API MediaStore oder den SDKs arbeiten oder diese anderweitig AWS-Services verwenden. AWS CLI AWS Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine

Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Datenverschlüsselung

MediaStore verschlüsselt Container und Objekte im Ruhezustand mithilfe des branchenüblichen AES-256-Algorithmus. Wir empfehlen Ihnen, Ihre Daten MediaStore auf folgende Weise zu sichern:

- Erstellen Sie eine Container-Richtlinie, um die Zugriffsrechte auf alle Ordner und Objekte in diesem Container zu kontrollieren. Weitere Informationen finden Sie unter the section called “Containerrichtlinien”.
- Erstellen Sie eine CORS-Richtlinie (Cross-Origin Resource Sharing), um den quellenübergreifenden Zugriff auf Ihre Ressourcen zu ermöglichen. MediaStore Mit CORS können Sie Client-Webanwendungen, die in einer Domain geladen sind, die Interaktion mit Ressourcen in einer anderen Domain erlauben. Weitere Informationen finden Sie unter the section called “CORS-Richtlinien”.

Identity and Access Management für AWS Elemental MediaStore

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Ressourcen zu verwenden. MediaStore IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

Zielgruppe

Die Art und Weise, wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, in der Sie tätig sind. MediaStore

Dienstbenutzer — Wenn Sie den MediaStore Dienst für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die erforderlichen Anmeldeinformationen und Berechtigungen zur Verfügung. Wenn Sie für Ihre Arbeit mehr MediaStore Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Fuktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anzufordern müssen. Wenn Sie in nicht auf eine Funktion zugreifen können MediaStore, finden Sie weitere Informationen unterFehlerbehebung bei AWS Elemental MediaStore Identity und Access.

Serviceadministrator — Wenn Sie in Ihrem Unternehmen für MediaStore Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf MediaStore. Es ist Ihre Aufgabe, zu bestimmen, auf welche MediaStore Funktionen und Ressourcen Ihre Servicebenutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen darüber, wie Ihr Unternehmen IAM nutzen kann MediaStore, finden Sie unterSo MediaStore funktioniert AWS Elemental mit IAM.

IAM-Administrator — Wenn Sie ein IAM-Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien schreiben können, um den Zugriff darauf zu verwalten. MediaStore Beispiele für MediaStore identitätsbasierte Richtlinien, die Sie in IAM verwenden können, finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS Elemental MediaStore

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS , übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter So melden Sie sich bei Ihrem an AWS-Konto im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, um Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu signieren, finden Sie im IAM-Benutzerhandbuch unter AWS API-Anfragen signieren.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS Empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter Multi-Faktor-Authentifizierung im AWS IAM Identity Center - Benutzerhandbuch und Verwenden der Multi-Faktor-Authentifizierung (MFA) in AWS im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Der Zugriff erfolgt, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center - Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Eine IAM-Gruppe ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter Erstellen eines IAM-Benutzers (anstatt einer Rolle) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der übernehmen, AWS Management Console indem Sie die Rollen wechseln. Sie können eine Rolle übernehmen, indem Sie eine AWS CLI oder AWS API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter Verwenden von IAM-Rollen im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Verbundbenutzerzugriff – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter Erstellen von Rollen für externe Identitätsanbieter im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter Berechtigungssätze im AWS IAM Identity Center -Benutzerhandbuch.
- Temporäre IAM-Benutzerberechtigungen – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- Kontoübergreifender Zugriff – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.
- Serviceübergreifender Zugriff — Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon-EC2 aus oder speichert Objekte in Amazon-S3. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Forward Access Sessions (FAS) — Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

- Servicerolle – Eine Servicerolle ist eine IAM-Rolle, die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.
- Dienstbezogene Rolle — Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

- Auf Amazon EC2 ausgeführte Anwendungen — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und API-Anfragen stellen AWS CLI . AWS Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2-Instance eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instance-Profil, das an die Instance angehängt ist. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon-EC2-Instances ausgeführt werden im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter Erstellen einer IAM-Rolle (anstatt eines Benutzers) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der AWS Management Console AWS CLI, der oder der AWS API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter Auswahl zwischen verwalteten und eingebundenen Richtlinien im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Services, die ACLs unterstützen. AWS WAF Weitere Informationen“ zu ACLs finden Sie unter Zugriffskontrollliste (ACL) – Übersicht (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- Berechtigungsgrenzen – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.
- Service Control Policies (SCPs) — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen. AWS Organizations AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Objekte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Entitäten. Root-Benutzer des AWS-Kontos Weitere Informationen zu Organizations und SCPs finden Sie unter Funktionsweise von SCPs im AWS Organizations -Benutzerhandbuch.

- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter Bewertungslogik für Richtlinien.

So MediaStore funktioniert AWS Elemental mit IAM

Bevor Sie IAM zur Verwaltung des Zugriffs auf verwenden, sollten Sie sich darüber informieren MediaStore, mit welchen IAM-Funktionen Sie arbeiten können. MediaStore

IAM-Funktionen, die Sie mit AWS Elemental verwenden können MediaStore

IAM-Feature	MediaStore Unterstützung
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Ja
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (services pezifisch)	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Teilweise

IAM-Feature	MediaStore Unterstützung
Temporäre Anmeldeinformationen	Ja
Hauptberechtigungen	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Nein

Einen allgemeinen Überblick darüber, wie MediaStore und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren.

Identitätsbasierte Richtlinien für MediaStore

Unterstützt Richtlinien auf Identitätsbasis.	Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für MediaStore

Beispiele für MediaStore identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS Elemental MediaStore

Ressourcenbasierte Richtlinien finden Sie in MediaStore

Unterstützt ressourcenbasierte Richtlinien	Ja

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM-Administrator des vertrauenswürdigen Kontos auch der Prinzipalentität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter Wie sich IAM-Rollen von ressourcenbasierten Richtlinien unterscheiden im IAM-Benutzerhandbuch.

> **Note**
>
> MediaStore unterstützt auch Container-Richtlinien, die definieren, welche Prinzipalentitäten (Konten, Benutzer, Rollen und Verbundbenutzer) Aktionen für den Container ausführen können. Weitere Informationen finden Sie unter Containerrichtlinien.

Richtlinienaktionen für MediaStore

Unterstützt Richtlinienaktionen	Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der MediaStore Aktionen finden Sie unter Von AWS Elemental definierte Aktionen MediaStore in der Service Authorization Reference.

Bei Richtlinienaktionen wird vor der Aktion das folgende Präfix MediaStore verwendet:

```
mediastore
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

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

Beispiele für MediaStore identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS Elemental MediaStore

Politische Ressourcen für MediaStore

Unterstützt Richtlinienressourcen	Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein – `Resource`oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

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

Eine Liste der MediaStore Ressourcentypen und ihrer ARNs finden Sie unter Von AWS Elemental definierte Ressourcen MediaStore in der Service Authorization Reference. Informationen zu den Aktionen, mit denen Sie den ARN jeder Ressource angeben können, finden Sie unter Von AWS Elemental MediaStore definierte Aktionen.

Die MediaStore Container-Ressource hat den folgenden ARN:

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

Weitere Informationen zum Format von ARNs finden Sie unter Amazon Resource Names (ARNs) und AWS Service Namespaces.

Um beispielsweise den Container `AwardsShow` in Ihrer Anweisung anzugeben, verwenden Sie den folgenden ARN:

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

Schlüssel zur Richtlinienbedingung für MediaStore

Unterstützt servicespezifische Richtlinienbedingungsschlüssel	Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen `AND`-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen `OR` Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

Eine Liste der MediaStore Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für AWS Elemental MediaStore in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von AWS Elemental MediaStore definierte Aktionen.

Beispiele für MediaStore identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS Elemental MediaStore

ACLs in MediaStore

Unterstützt ACLs	Nein

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit MediaStore

Unterstützt ABAC (Tags in Richtlinien)	Teilweise

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In AWS werden diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Was ist ABAC? im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

Verwenden temporärer Anmeldeinformationen mit MediaStore

Unterstützt temporäre Anmeldeinformationen	Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services funktionieren AWS-Services , finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn

Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter Wechseln zu einer Rolle (Konsole) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldeinformationen in IAM.

Serviceübergreifende Prinzipalberechtigungen für MediaStore

Unterstützt Forward Access Sessions (FAS)	Ja

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für MediaStore

Unterstützt Servicerollen	Ja

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

> **Warning**
>
> Durch das Ändern der Berechtigungen für eine Servicerolle kann die MediaStore Funktionalität beeinträchtigt werden. Bearbeiten Sie Servicerollen nur, MediaStore wenn Sie dazu eine Anleitung erhalten.

Dienstbezogene Rollen für MediaStore

Unterstützt serviceverknüpfte Rollen	Nein

Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie in der Tabelle nach einem Service mit einem `Yes` in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Beispiele für identitätsbasierte Richtlinien für AWS Elemental MediaStore

Standardmäßig sind Benutzer und Rollen nicht berechtigt, Ressourcen zu erstellen oder zu ändern. MediaStore Sie können auch keine Aufgaben mithilfe der AWS Management Console, AWS Command Line Interface (AWS CLI) oder AWS API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Einzelheiten zu Aktionen und Ressourcentypen, die von definiert wurden MediaStore, einschließlich des Formats der ARNs für jeden der Ressourcentypen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS Elemental MediaStore in der Service Authorization Reference.

Themen

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand MediaStore Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um Ihren Benutzern und Workloads zunächst Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter AWS -verwaltete Richtlinien oder AWS -verwaltete Richtlinien für Auftrags-Funktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.

- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung zum IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Konfigurieren eines MFA-geschützten API-Zugriffs im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der MediaStore-Konsole

Um auf die AWS Elemental MediaStore Console zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, die MediaStore Ressourcen in Ihrem AWS-Konto aufzulisten und Details zu diesen anzuzeigen. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen die MediaStore Konsole weiterhin verwenden können, fügen Sie den Entitäten auch die MediaStore `ConsoleAccess` oder die `ReadOnly` AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zu einem Benutzer im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer

Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der AWS CLI AWS OR-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": "*"
        }
    ]
}
```

Fehlerbehebung bei AWS Elemental MediaStore Identity und Access

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit MediaStore und IAM auftreten können.

Themen

Ich bin nicht berechtigt, eine Aktion durchzuführen in MediaStore

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven *`my-example-widget`*-Ressource anzuzeigen, jedoch nicht über `mediastore:`*`GetWidget`*-Berechtigungen verfügt.

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

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `mediastore:`*`GetWidget`*-Aktion auf die *`my-example-widget`*-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam durchzuführen: PassRole

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht berechtigt sind, die `iam:PassRole` Aktion auszuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an MediaStore diese Person übergeben können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in auszuführen. MediaStore Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

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

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine MediaStore Ressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob diese Funktionen MediaStore unterstützt werden, finden Sie unterSo MediaStore funktioniert AWS Elemental mit IAM.
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.

Anmeldung und Überwachung AWS Elemental MediaStore

Dieser Abschnitt bietet eine Übersicht über die Optionen zur Protokollierung und Überwachung in AWS Elemental MediaStore zu Sicherheitszwecken. Weitere Informationen zur Anmeldung und Überwachung finden Sie unterÜberwachung und Tagging in AWS Elemental MediaStore. MediaStore

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit AWS Elemental MediaStore und Leistung Ihrer AWS Lösungen. Sie sollten Überwachungsdaten aus allen Teilen Ihrer AWS Lösung sammeln, damit Sie einen etwaigen Ausfall an mehreren Stellen leichter debuggen können. AWS bietet mehrere Tools zur Überwachung Ihrer MediaStore Ressourcen und zur Reaktion auf potenzielle Vorfälle.

CloudWatch Amazon-Alarme

Mithilfe von CloudWatch Alarmen beobachten Sie eine einzelne Metrik über einen von Ihnen festgelegten Zeitraum. Wenn die Metrik einen bestimmten Schwellenwert überschreitet, wird eine Benachrichtigung an ein Amazon SNS SNS-Thema oder eine AWS Auto Scaling Scaling-Richtlinie gesendet. CloudWatch Alarme lösen keine Aktionen aus, da sie sich in einem bestimmten Status befinden. Der Status muss sich stattdessen geändert haben und für eine festgelegte Anzahl an Zeiträumen aufrechterhalten worden sein. Weitere Informationen finden Sie unter Überwachung mit CloudWatch.

AWS CloudTrail Logs

CloudTrail bietet eine Aufzeichnung der Aktionen, die von einem Benutzer, einer Rolle oder einem AWS Dienst in ausgeführt wurden AWS Elemental MediaStore. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, an die die Anfrage gestellt wurde MediaStore, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Informationen ermitteln. Weitere Informationen finden Sie unter Protokollierung von API-Aufrufen mit CloudTrail.

AWS Trusted Advisor

Trusted Advisor stützt sich auf bewährte Verfahren, die wir bei der Betreuung von Hunderttausenden von AWS Kunden gelernt haben. Trusted Advisor untersucht Ihre AWS-Umgebung und gibt dann Empfehlungen, wenn Möglichkeiten bestehen, Geld zu sparen, die Systemverfügbarkeit und -leistung zu verbessern oder Sicherheitslücken zu schließen. Alle AWS Kunden haben Zugriff auf fünf Trusted Advisor Advisor-Checks. Kunden mit einem Business- oder Enterprise-Supportplan können alle Trusted Advisor Schecks einsehen.

Weitere Informationen finden Sie unter AWS Trusted Advisor.

Konformitätsvalidierung für AWS Elemental MediaStore

Informationen darüber, ob AWS-Service ein AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter Umfang nach Compliance-Programm AWS-Services unter . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter AWS Compliance-Programme AWS .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- Schnellstartanleitungen zu Sicherheit und Compliance — In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Implementierung von Basisumgebungen beschrieben AWS , bei denen Sicherheit und Compliance im Mittelpunkt stehen.
- Architecting for HIPAA Security and Compliance on Amazon Web Services — In diesem Whitepaper wird beschrieben, wie Unternehmen HIPAA-fähige Anwendungen erstellen AWS können.

> **Note**
>
> AWS-Services Nicht alle sind HIPAA-fähig. Weitere Informationen finden Sie in der Referenz für HIPAA-berechtigte Services.

- AWS Compliance-Ressourcen — Diese Sammlung von Arbeitsmappen und Leitfäden gilt möglicherweise für Ihre Branche und Ihren Standort.
- AWS Leitfäden zur Einhaltung von Vorschriften für Kunden — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services und die Leitlinien den Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zugeordnet.

- Evaluierung von Ressourcen anhand von Regeln im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub— Dies AWS-Service bietet einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der Security-Hub-Steuerungsreferenz.
- Amazon GuardDuty — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen wie PCI DSS zu erfüllen, indem es die in bestimmten Compliance-Frameworks vorgeschriebenen Anforderungen zur Erkennung von Eindringlingen erfüllt.
- AWS Audit Manager— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Resilienz in AWS Elemental MediaStore

Die AWS globale Infrastruktur basiert AWS-Regionen auf Availability Zones. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu Availability Zones AWS-Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Zusätzlich zur AWS globalen Infrastruktur MediaStore bietet es mehrere Funktionen, die Sie bei Ihren Anforderungen an Datenstabilität und Datensicherung unterstützen.

Infrastruktursicherheit in AWS Elemental MediaStore

Als verwalteter Service MediaStore ist AWS Elemental durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsservices und zum AWS Schutz der Infrastruktur

finden Sie unter AWS Cloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe für den Zugriff MediaStore über das Netzwerk. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Serviceübergreifende Confused-Deputy-Prävention

Das Confused-Deputy-Problem ist ein Sicherheitsproblem, bei dem eine juristische Stelle, die nicht über die Berechtigung zum Ausführen einer Aktion verfügt, eine privilegiertere juristische Stelle zwingen kann, die Aktion auszuführen. Im AWS Fall eines dienstübergreifenden Identitätswechsels kann das Problem des verwirrten Stellvertreters auftreten. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der aufrufende Service kann manipuliert werden, um seine Berechtigungen zu verwenden, um Aktionen auf die Ressourcen eines anderen Kunden auszuführen, für die er sonst keine Zugriffsberechtigung haben sollte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Wir empfehlen, die Kontextschlüssel `aws:SourceArn`und die `aws:SourceAccount`globalen Bedingungsschlüssel in Ressourcenrichtlinien zu verwenden, um die Berechtigungen einzuschränken, die AWS Elemental einem anderen Service für die Ressource MediaStore erteilt. Verwenden Sie `aws:SourceArn`, wenn Sie nur eine Ressource mit dem betriebsübergreifenden Zugriff verknüpfen möchten. Verwenden Sie `aws:SourceAccount`, wenn Sie zulassen möchten, dass Ressourcen in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft werden.

Der effektivste Weg, um sich vor dem Confused-Deputy-Problem zu schützen, ist die Verwendung des globalen Bedingungskontext-Schlüssels `aws:SourceArn` mit dem vollständigen ARN der Ressource. Wenn Sie den vollständigen ARN der Ressource nicht kennen oder wenn Sie mehrere Ressourcen angeben, verwenden Sie den globalen Kontextbedingungsschlüssel `aws:SourceArn` mit Platzhalterzeichen (*) für die unbekannten Teile des ARN. z. B. `arn:aws:servicename:*:123456789012:*`.

Wenn der `aws:SourceArn`-Wert die Konto-ID nicht enthält, z. B. einen Amazon-S3-Bucket-ARN, müssen Sie beide globale Bedingungskontextschlüssel verwenden, um Berechtigungen einzuschränken.

Der Wert von `aws:SourceArn` muss die Konfiguration sein, für die CloudWatch Protokolle in Ihrer Region und Ihrem Konto MediaStore veröffentlicht werden.

Das folgende Beispiel zeigt, wie Sie die Kontextschlüssel `aws:SourceArn` und die `aws:SourceAccount` globale Bedingung verwenden können, MediaStore um das Problem des verwirrten Stellvertreters zu vermeiden.

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

Überwachung und Tagging in AWS Elemental MediaStore

Überwachung ist wichtig, um die Zuverlässigkeit, Verfügbarkeit und Leistung von AWS Elemental MediaStore und Ihrer anderenAWS -Lösungen aufrechtzuerhalten. AWSstellt die folgenden Überwachungstools bereit, um zu überwachen MediaStore, zu melden, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen zu ergreifen:

- AWS CloudTrail erfasst API-Aufrufe und zugehörige Ereignisse, die von oder im Namen Ihres AWS-Kontos erfolgten, und übermittelt die Protokolldateien an einen von Ihnen angegebenen Amazon-S3-Bucket. Sie können die Benutzer und Konten, die AWS aufgerufen haben, identifizieren, sowie die Quell-IP-Adresse, von der diese Aufrufe stammen, und den Zeitpunkt der Aufrufe ermitteln. Weitere Informationen finden Sie im AWS CloudTrail-Benutzerhandbuch.
- Amazon CloudWatch überwacht IhreAWS -Ressourcen und dieAWS in ausgeführten Anwendungen in Echtzeit. Sie können Metriken erfassen und verfolgen, benutzerdefinierte Dashboards erstellen und Alarme festlegen, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen von Ihnen festgelegten Schwellenwert erreicht. Beispielsweise können Sie mit die CPU-Auslastung oder andere Metriken Ihrer Amazon EC2-Instances CloudWatch erfassen und bei Bedarf automatisch neue Instances starten. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.
- Amazon CloudWatch Events liefert einen Strom von Systemereignissen, die Änderungen inAWS -Ressourcen beschreiben. Ereignisbenachrichtigungen stellenAWS -Services typischerweise in wenigen Sekunden bereit CloudWatch , manchmal kann dies aber auch eine Minute oder länger dauern. CloudWatch Events ermöglicht automatisierte, ereignisgesteuerte Datenverarbeitung, denn Sie können Regeln schreiben, die bestimmte Ereignisse überwachen und automatisierte Aktionen in anderenAWS -Services auslösen, wenn diese Ereignisse auftreten. Weitere Informationen finden Sie im Amazon CloudWatch Events-Benutzerhandbuch.
- Amazon CloudWatch Logs ermöglicht Ihnen die Überwachung, Speicherung und den Zugriff auf Ihre Protokolldateien von Amazon EC2-Instances und anderen Quellen. CloudTrail CloudWatch Protokolle können Informationen in den Protokolldateien überwachen und Sie benachrichtigen, wenn bestimmte Schwellenwerte erreicht werden. Sie können Ihre Protokolldaten auch in einem sehr robusten Speicher archivieren. Weitere Informationen finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch.

Sie können Metadaten Ihren MediaStore Containern auch Metadaten in Form von Tags zuweisen. Jedes Tag ist eine Markierung, die aus einem von Ihnen definierten Schlüssel und Wert besteht.

Tags können die Verwaltung, Suche und Filterung von Ressourcen erleichtern. Sie können Tags verwenden, um Ihre AWS-Ressourcen in der AWS Management Console zu organisieren, um Nutzungs- und Fakturierungs-Berichte innerhalb aller Ihrer AWS-Ressourcen zu erstellen und um Ressourcen während der Infrastrukturautomatisierung zu filtern.

Themen

Protokollieren von AWS Elemental MediaStore API-Aufrufen mitAWS CloudTrail

AWS Elemental MediaStore ist in integriertAWS CloudTrail, einen Service, der die von einem Benutzer, einer Rolle oder einemAWS -Service in durchgeführten Aktionen erfasst MediaStore. CloudTrail erfasst eine Teilmenge von API-Aufrufen, die MediaStore als Ereignisse bereitgestellt werden, einschließlich Aufrufen von der MediaStore -Konsole und von Code-Aufrufen an die MediaStore -API. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignisse für MediaStore. Auch wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse in der CloudTrail -Konsole in Event history (Ereignisverlauf) anzeigen. Mit den von CloudTrail gesammelten Informationen können Sie die an gestellte Anfrage MediaStore, die IP-Adresse, von der die Anfrage gestellt wurde, den Initiator der Anfrage, den Zeitpunkt der Anfrage und mehr bestimmen.

Weitere Informationen CloudTrail, einschließlich Konfiguration und Aktivierung, finden Sie im AWS CloudTrail-Benutzerhandbuch.

Themen

MediaStoreInformationen zu AWS Elemental in CloudTrail

CloudTrail wird beim Erstellen IhresAWS -Kontos für Sie aktiviert. Die in AWS Elemental MediaStore auftretenden unterstützten Aktivitäten werden als CloudTrail Ereignis zusammen mit anderenAWS -Serviceereignissen im Ereignisverlauf aufgezeichnet. Sie können die neusten Ereignisse in Ihr AWS-

Konto herunterladen und dort suchen und anzeigen. Weitere Informationen finden Sie unter Anzeigen von Ereignissen mit dem CloudTrail -API-Ereignisverlauf.

Zur kontinuierlichen Aufzeichnung von Ereignissen in Ihrem AWS-Konto, einschließlich Ereignissen für MediaStore, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Bereitstellung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS-Partition und stellt die Protokolldateien in dem von Ihnen angegebenen Amazon S3 Bucket bereit. Darüber hinaus können Sie andereAWS -Services konfigurieren, um die in den CloudTrail -Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren. Weitere Informationen finden Sie unter den folgenden Themen:

- Übersicht zum Erstellen eines Trails
- CloudTrail Unterstützte Dienste und Integrationen
- Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten.

AWS Elemental MediaStore unterstützt die Protokollierung der folgenden Vorgänge als Ereignisse in CloudTrail Protokolldateien:

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

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Anmeldeinformationen des Root-Benutzers oder des -Benutzers gestellt wurde
- Ob die Anfrage mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen föderierten Benutzer ausgeführt wurde
- Ob die Anforderung von einem anderen AWS-Service getätigt wurde.

Weitere Informationen finden Sie unter CloudTrail userIdentity-Element.

Beispiel: AWS Elemental MediaStore Elemental-Protokolldateieinträge

Ein Trail ist eine Konfiguration, durch die Ereignisse als Protokolldateien an den von Ihnen angegebenen Amazon-S3-Bucket übermittelt werden. CloudTrail Protokolldateien können einen oder mehrere Einträge enthalten. Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält unter anderem Informationen über die angeforderte Aktion, das Datum und die Uhrzeit der Aktion sowie über die Anfrageparameter. CloudTrail Protokolldateien sind kein geordnetes Stacktrace der öffentlichen API-Aufrufe und erscheinen daher nicht in einer bestimmten Reihenfolge.

Das folgende Beispiel zeigt einen CloudTrail -Protokolleintrag, der die`CreateContainer` -Operation demonstriert:

```
{
    "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"
}
```


Überwachung von AWS Elemental MediaStore mit Amazon CloudWatch

Sie können AWS Elemental MediaStore mithilfe von Berechtigungen CloudWatch überwachen. Dabei werden Rohdaten gesammelt und zu lesbaren Metriken verarbeitet. CloudWatch speichert die Statistiken 15 Monate, damit Sie auf Verlaufsinformationen zugreifen können und einen besseren Überblick darüber erhalten, wie Ihre Webanwendung oder der Service ausgeführt wird. Sie können auch Alarme einrichten, die auf bestimmte Grenzwerte achten und Benachrichtigungen senden oder Aktivitäten auslösen, wenn diese Grenzwerte erreicht werden. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

AWSstellt die folgenden Überwachungstools bereit, um zu überwachen MediaStore, zu melden, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen zu ergreifen:

- Amazon CloudWatch Logs ermöglicht Ihnen die Überwachung, Speicherung und den Zugriff auf Ihre Protokolldateien vonAWS Diensten wie AWS Elemental MediaStore. Sie können CloudWatch Protokolle verwenden, um Anwendungen und Systeme, die Protokolldaten verwenden, zu überwachen. Mit CloudWatch Logs können Sie beispielsweise die Fehler zählen, die in Ihren Anwendungsprotokollen aufgeführt werden, und Ihnen eine Benachrichtigung senden, wenn die

Fehlerrate einen von Ihnen festgelegten Schwellenwert überschreitet. CloudWatch Logs verwendet Ihre Protokolldaten zur Überwachung, daher sind keine Code-Änderungen erforderlich. Sie können beispielsweise Anwendungsprotokolle auf bestimmte wörtliche Begriffe (wie "ValidationException„) überwachen oder die Anzahl der`PutObject` Anfragen zählen, die in einem bestimmten Zeitraum gestellt wurden. Wird der von Ihnen gesuchte Begriff gefunden, CloudWatch meldet Logs die Daten an eine von Ihnen angegebene CloudWatch Metrik. Die Protokolldaten werden während der Übermittlung und Speicherung verschlüsselt.

- Amazon CloudWatch Events liefert Systemereignisse, die Änderungen anAWS Ressourcen, z. B. MediaStore Objekten, beschreiben. Ereignisbenachrichtigungen stellenAWS -Services typischerweise in wenigen Sekunden bereit CloudWatch , manchmal kann dies aber auch eine Minute oder länger dauern. Sie können Regeln einrichten, die Ereignissen entsprechen (z. B. einer`DeleteObject` Anforderung) und sie an eine oder mehrere Zielfunktionen oder Streams umleiten. CloudWatch Ereignisse bemerkt betriebsbezogene Veränderungen, sobald diese auftreten. CloudWatch Events reagiert außerdem auf diese betriebsbezogenen Änderungen und führt bei Bedarf Korrekturmaßnahmen durch, indem es an die Umgebung Nachrichten versendet, Funktionen aktiviert, Änderungen vornimmt und Zustandsinformationen erfasst.

CloudWatch Logs

Die Zugriffsprotokollierung stellt detaillierte Aufzeichnungen über die Anfragen bereit, die an Objekte in einem Container gestellt wurden. Zugriffsprotokolle sind für für viele Anwendungen nützlich, wie z. B. für Sicherheits- und Zugriffsüberprüfungen. Sie können Ihnen auch dabei helfen, mehr über Ihren Kundenstamm zu erfahren und Ihre MediaStore -Rechnung zu verstehen. CloudWatch Protokolle sind wie folgt kategorisiert:

- Ein Protokollstream ist eine Abfolge von Protokollereignissen, die dieselbe Quelle nutzen.
- Eine Protokollgruppe ist eine Gruppe von Protokollstreams, die dieselben Einstellungen für die Aufbewahrung, Überwachung und Zugriffskontrolle besitzen. Wenn Sie die Zugriffsprotokollierung für einen Container aktivieren, MediaStore wird eine Protokollgruppe mit einem Namen wie erstellt`/aws/mediastore/MyContainerName`. Sie können Protokollgruppen definieren und angeben, welche Streams in welche Gruppe geschickt werden sollen. Es gibt kein Kontingent dazu, wie viele Protokoll-Streams zu einer Protokollgruppe gehören können.

Standardmäßig werden Protokolle unbegrenzt aufbewahrt und laufen nicht ab. Sie können die Aufbewahrungsrichtlinie für jede Protokollgruppe anpassen und Protokolle entweder unbegrenzt speichern oder einen Aufbewahrungszeitraum zwischen einem Tag und 10 Jahren auswählen.

Einrichten von Berechtigungen für Amazon von Berechtigungen von Berechtigungen für Amazon CloudWatch

Verwenden SieAWS Identity and Access Management (IAM), um eine Rolle zu erstellen, die AWS Elemental MediaStore Zugriff auf Amazon gewährt CloudWatch. Sie müssen diese Schritte ausführen, damit die CloudWatch Protokolle für Ihr Konto veröffentlicht werden. CloudWatch veröffentlicht automatisch Kennzahlen für Ihr Konto.

Um den MediaStore Zugriff auf zu ermöglichen CloudWatch

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie in der IAM-Konsole im Navigationsbereich die Option Policies (Richtlinien) und dann Create policy (Richtlinie erstellen) aus.
3. Wählen Sie die Registerkarte JSON und fügen Sie dann die folgende Richtlinie ein:

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

Diese Richtlinie ermöglicht MediaStore die Erstellung von Protokollgruppen und Log-Streams für alle Container in jeder Region innerhalb IhresAWS Kontos.

4. Wählen Sie Review policy (Richtlinie prüfen).
5. Geben Sie auf der Seite Review policy (Richtlinie prüfen) für Name den Namen **`MediaStoreAccessLogsPolicy`** ein und wählen Sie dann Create policy (Richtlinie erstellen).
6. Klicken Sie im Navigationsbereich der IAM-Konsole auf Roles und wählen Sie dann Create role.
7. Wählen Sie den Rollentyp Another AWS account (Anderes AWS-Konto) aus.
8. Geben Sie für Account ID (Konto-ID) Ihre AWS-Konto-ID ein.
9. Wählen Sie Next: Permissions (Weiter: Berechtigungen) aus.
10. Geben Sie in das Suchfeld ei **`MediaStoreAccessLogsPolicy`**.
11. Aktivieren Sie das Kontrollkästchen neben der neuen Richtlinie und klicken Sie dann auf Next: Tags (Weiter: Tags).
12. Wählen Sie Next: Review (Weiter: Prüfen) aus, um eine Vorschau Ihres neuen Benutzers anzuzeigen.
13. Geben Sie für Role name (Rollenname) den Namen **`MediaStoreAccessLogs`** ein und klicken Sie auf Create role (Rolle erstellen).
14. Wählen Sie in der Bestätigungsmeldung den Namen der Rolle, die Sie gerade erstellt haben (**`MediaStoreAccessLogs`**).
15. Wählen Sie auf der Seite Summary (Übersicht) der Rolle die Registerkarte Trust relationships (Vertrauensstellungen).
16. Wählen Sie Edit Trust Relationship (Vertrauensstellungen bearbeiten).
17. Ändern Sie im Richtliniendokument den Prinzipal auf den MediaStore-Service. Das sollte wie folgt aussehen:

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

Die gesamte Richtlinie sollte folgendermaßen lauten:

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

```
      "Action": "sts:AssumeRole",
      "Condition": {}
    }
  ]
}
```

18. Wählen Sie Update Trust Policy (Trust Policy aktualisieren).

Aktivieren der Zugriffsprotokollierung für einen Container

Standardmäßig erfasst AWS Elemental MediaStore keine Zugriffsprotokolle. Wenn Sie die Zugriffsprotokollierung eines Containers aktivieren MediaStore , werden Zugriffsprotokolle für in diesem Container gespeicherte Objekte an Amazon gesendet CloudWatch. Die Zugriffsprotokolle enthalten detaillierte Datensätze für Anfragen, die an ein beliebiges im Container gespeichertes Objekt gestellt wurden. Dabei kann es sich um den Anforderungstyp, die in der Anfrage angegebenen Ressourcen sowie Uhrzeit und Datum der Anfrageverarbeitung handeln.

> **Important**
>
> Für die Aktivierung der Zugriffsprotokollierung auf einem MediaStore-Container fallen keine zusätzlichen Kosten an. Für die Protokolldateien, die der Service an Sie überträgt, fallen die normalen Gebühren für die Speicherung an. (Sie können die Protokolldateien jederzeit löschen.) AWS berechnet keine Datenübertragungskosten für die Übertragung der Protokolldateien. Für den Zugriff auf die Protokolldateien fällt aber die normale Gebühr für Datenübertragungen an.

So aktivieren Sie die Zugriffsprotokollierung (AWS CLI):

- Verwenden Sie in der AWS CLI den Befehl `start-access-logging`:

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

  Dieser Befehl hat keinen Rückgabewert.

Deaktivieren der Zugriffsprotokollierung für einen Container

Wenn Sie die Zugriffsprotokollierung für einen Container deaktivieren, sendet AWS Elemental MediaStore keine Zugriffsprotokolle mehr an Amazon CloudWatch. Diese Zugriffsprotokolle werden nicht gespeichert und können nicht abgerufen werden.

Deaktivieren der Zugriffsprotokollierung (AWS CLI)

- Verwenden Sie in der AWS CLI den Befehl `stop-access-logging`:

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

  Dieser Befehl hat keinen Rückgabewert.

Fehlerbehebung bei der Zugriffsprotokollierung in AWS Elemental MediaStore

Wenn die MediaStore Zugriffsprotokolle von AWS Elemental in Amazon nicht angezeigt werden CloudWatch, finden Sie in der folgenden Tabelle mögliche Ursachen und Lösungen.

> **Note**
>
> Aktivieren Sie die AWS CloudTrail-Protokolle zur Unterstützung bei der Fehlerbehebung.

Symptom	Das Problem ist möglicherweise...	Versuchen Sie...
Sie sehen keine CloudTrail Ereignisse, obwohl CloudTrail Protokolle aktiviert sind.	Die IAM-Rolle ist entweder nicht vorhanden oder hat einen falschen Namen, falsche Berechtigungen oder eine falsche Vertrauensrichtlinie.	Erstellen Sie eine Rolle mit dem richtigen Namen, den richtigen Berechtigungen und der richtigen Vertrauensrichtlinie. Siehe the section called “Einrichten von Berechtigungen für die Einrichten von Berechtigungen für CloudWatch”.

Symptom	Das Problem ist möglicherweise...	Versuchen Sie...
Sie haben eine `DescribeContainer` -API-Anfrage gestellt, aber die Antwort zeigt, dass der `AccessLoggingEnabled` -Parameter den Wert `False` aufweist. Außerdem können Sie keine CloudTrail-Ereignisse für die `MediaStoreAccessLogs` -Rolle sehen, die einen erfolgreichen `DescribeLogGroup` -, `CreateLogGroup` -, `DescribeLogStream` - oder `CreateLogStream` -Anruf tätigt.	Die IAM-Rolle ist entweder nicht vorhanden oder hat einen falschen Namen, falsche Berechtigungen oder eine falsche Vertrauensrichtlinie.	Erstellen Sie eine Rolle mit dem richtigen Namen, den richtigen Berechtigungen und der richtigen Vertrauensrichtlinie. Siehe the section called “Einrichten von Berechtigungen für die Einrichten von Berechtigungen für CloudWatch”.
	Die Zugriffsprotokollierung ist auf dem Container nicht aktiviert.	Aktivieren Sie die Zugriffsprotokolle für den Container . Siehe the section called “Aktivieren der Zugriffsprotokollierung”.

Symptom	Das Problem ist möglicherweise...	Versuchen Sie...

Auf der CloudTrail Konsole wird ein Ereignis mit dem Fehler „Zugriff verweigert“ im Zusammenhang mit der`MediaStoreAccessLogs` Rolle angezeigt. Das CloudTrail Ereignis kann Zeilen wie die folgenden enthalten :
 	
`"eventSource": "logs.amazonaws.com",`
 	
`"errorCode": "AccessDenied",`
 	
`"errorMessage": "User: arn:aws:sts::111122223333:assumed-role/MediaStoreAccessLogs/MediaStoreAccessLogsSession is not authorized to perform: logs:DescribeLogGroups on resource: arn:aws:logs:us-west-2:111122223333:log-group::log-stream:",`
	Die IAM-Rolle hat nicht die richtigen Berechtigungen für AWS Elemental MediaStore.	Aktualisieren Sie die IAM-Rolle mit den richtigen Berechtigungen und Vertrauen srichtlinien. Siehe the section called “Einrichten von Berechtigungen für die Einrichten von Berechtigungen für CloudWatch”.

Symptom	Das Problem ist möglicherweise...	Versuchen Sie...
Sie sehen keine Protokolle für einen ganzen Container oder mehrere Container.	Ihr Konto hat möglicherweise das CloudWatch Kontingent für Protokollgruppen pro Region überschritten. Die Kontingente für Protokollgruppen finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch.	Stellen Sie auf der CloudWatch Konsole fest, ob Ihr Konto das CloudWatch Kontingent für Protokollgruppen erreicht hat. Bei Bedarf fordern Sie eine Kontingenterhöhung an.
Sie sehen einige Logins CloudWatch, aber nicht alle Logs, die Sie erwarten.	Ihr Konto hat möglicherweise das CloudWatch Kontingent für Transaktionen pro Sekunde pro Konto pro Region überschritten. Die Kontingente für finden Sie`PutLogEvents` im Amazon CloudWatch Logs-Benutzerhandbuch.	Beantragen Sie eine Kontingenterhöhung für CloudWatch Transaktionen pro Sekunde pro Sekunde und Konto und Region.

Zugriffsprotokollformat

Die Zugriffsprotokolldateien bestehen aus einer Reihe von JSON-formatierten Protokolldatensätzen, wobei jeder Protokolldatensatz eine Anfrage darstellt. Die Reihenfolge der Felder innerhalb des Protokolls kann variieren. Im Folgenden finden Sie ein Beispielprotokoll, das aus zwei Protokolldatensätzen besteht:

```
{
```

```
  "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
}
```

In der folgenden Liste werden die wichtigsten Protokolldatensatzfelder beschrieben:

AWSAccountId

Die AWS-Konto-ID des Kontos, das zum Erstellen der Anfrage verwendet wurde.

BytesReceived

Die Anzahl der Bytes im Anforderungstext, die der MediaStore-Server empfängt.

BytesSent

Die Anzahl der Bytes im Antworttext, die der MediaStore-Server sendet. Dieser Wert ist häufig identisch mit dem Wert des `Content-Length`-Headers, der in Serverantworten enthalten ist.

ContainerName

Der Name des Containers, der die Anfrage empfangen hat.

ErrorCode

Der MediaStore Fehlercode (z. B.`InternalServerError`). Wenn keine Fehler aufgetreten, wird das Zeichen `-` angezeigt. Ein Fehlercode kann sogar angezeigt werden, wenn der Statuscode 200 lautet (wird angezeigt, wenn eine geschlossene Verbindung oder eine Fehlermeldung angezeigt wird, nachdem der Server mit dem Streamen der Antwort begonnen hat).

ExpiresAt

Datum und Uhrzeit, an dem das Objekt abläuft. Dieser Wert basiert auf dem Verfallsdatum, das durch einetransient data rule im Lebenszyklus enthaltene Richtlinie festgelegt wurde, die auf den Container angewendet wird. Der Wert ist `ISO-8601`-Datum und -Uhrzeit und basiert auf der Systemuhr des Hosts, der die Anfrage verarbeitet. Wenn die Lebenszyklusrichtlinie keine transiente Datenregel enthält, die für das Objekt gilt, oder wenn keine Lebenszyklusrichtlinie auf den Container angewendet wird, lautet der Wert dieses Feldsnull. Dieses Feld gilt nur für die folgenden Operationen:`PutObject`,`GetObject`,`DescribeObject`, und`DeleteObject`.

HTTPStatus

Der numerische HTTP-Statuscode der Antwort.

Operation

Die Operation, die durchgeführt wurde, z. B. `PutObject` oder `ListItems`.

Pfad

Der Pfad im Container, in dem das Objekt gespeichert ist. Wenn die Operation keinen Pfadparameter verwendet, wird das Zeichen `-` angezeigt.

ReceivedTime

Die Tageszeit, zu der die Anfrage empfangen wurde. Der Wert ist `ISO-8601`-Datum und -Uhrzeit und basiert auf der Systemuhr des Hosts, der die Anfrage verarbeitet.

Auftraggeber

Der Amazon Resource Name (ARN) des Benutzers des Kontos, das zum Erstellen der Anfrage verwendet wurde. Bei nicht authentifizierten Anfragen ist dieser Wert `anonymous`. Wenn die Anforderung fehlschlägt, bevor die Authentifizierung abgeschlossen ist, fehlt dieses Feld möglicherweise im Protokoll. Bei solchen Anforderungen ist möglicherweise am `ErrorCode` das Autorisierungsproblem zu erkennen.

RequestID

Eine Zeichenfolge, die von AWS Elemental generiert wird MediaStore , um jede Anforderung eindeutig zu identifizieren.

Quelle

Die offensichtliche Internetadresse des Auftraggebers oder des Service-Prinzipals des AWS-Service, der den Aufruf vornimmt. Wenn zwischengeschaltete Proxys und Firewalls die Adresse des Computers verschleiern, der die Anfrage stellt, wird der Wert auf Null gesetzt.

TotalTime

Die Anzahl der Millisekunden (ms), die die Anfrage aus Perspektive des Servers unterwegs war. Dieser Wert wird ab der Zeit gemessen, zu der Ihre Anfrage vom Service empfangen wird, und bis zu der Zeit, zu der das letzte Byte der Antwort gesendet wurde. Dieser Wert wird von der Perspektive des Servers aus gemessen, da Messungen aus der Perspektive des Clients von der Netzwerklatenz beeinträchtigt sind.

TurnAroundTime

Die Anzahl der Millisekunden, die MediaStore Ihre Anfrage verarbeitet hat. Dieser Wert wird ab der Zeit gemessen, zu der das letzte Byte Ihrer Anforderung empfangen wurde, bis zu der Zeit, zu der das erste Byte der Antwort gesendet wurde.

Die Reihenfolge der Felder in der Protokolldatei kann variieren.

Protokollierungsstatusänderungen werden mit der Zeit wirksam.

Änderungen am Protokollierungsstatus eines Containers benötigen einige Zeit, bis sie sich auf die Bereitstellung von Protokolldateien auswirken. Wenn Sie beispielsweise die Protokollierung für einen Container A aktivieren, werden möglicherweise einige Anfragen, die in der darauffolgenden Stunde gestellt werden, protokolliert, andere hingegen nicht. Wenn Sie die Protokollierung für Container

B deaktivieren, werden in der nächsten Stunde einige Protokolle möglicherweise weiter zugestellt, andere möglicherweise nicht. Die neuen Einstellungen werden letztendlich in allen Fällen ohne weiteres Eingreifen Ihrerseits wirksam.

Best-Effort-Protokollbereitstellung der Server

Zugriffsprotokoll-Datensätze werden auf Best-Effort-Basis bereitgestellt. Die meisten Anfragen nach einem Container, der für die Protokollierung richtig konfiguriert ist, führen zu einem ausgelieferten Protokollsatz. Die meisten Protokollsätze werden innerhalb weniger Stunden nach der Aufnahme geliefert, können aber häufiger geliefert werden.

Die Vollständigkeit und Aktualität der Zugriffsprotokollierung wird nicht garantiert. Der Protokolldatensatz für eine bestimmte Anforderung wird möglicherweise viel später bereitgestellt, als die Anforderung tatsächlich verarbeitet wurde; es kann auch sein, dass er gar nicht bereitgestellt wird. Der Zweck der Zugriffsprotokolle besteht darin, Ihnen einen Überblick über die Art des Datenverkehrs zu und von Ihrem Container zu vermitteln. Es passiert selten, das Protokolldatensätze verloren gehen, aber die Zugriffsprotokollierung ist nicht als vollständige Auflistung aller Anfragen vorgesehen.

Aufgrund der Best-Effort-Natur der Zugriffsprotokollierungsfunktion können die im AWS-Portal verfügbaren Nutzungsberichte (Abrechnungs- und Kostenverwaltungsberichte auf der AWS Management Console) eine oder mehrere Zugriffsanfragen enthalten, die nicht in einem bereitgestellten Zugriffsprotokoll angezeigt werden.

Überlegungen bei der Programmierung des Zugriffsprotokollformats

Wir erweitern möglicherweise von Zeit zu Zeit das Zugriffsprotokollformat, indem wir neue Felder hinzufügen. Code, der Zugriffprotokolle analysiert, muss so geschrieben werden, dass er zusätzliche Felder verarbeiten kann, die er nicht versteht.

CloudWatch Ereignisse

Mit Amazon CloudWatch Events können Sie IhreAWS -Services automatisieren und automatisch auf Systemereignisse reagieren, z. B. bei Problemen mit der Anwendungsverfügbarkeit oder Ressourcenänderungen. Sie können einfache Regeln schreiben, um anzugeben, welche Ereignisse für Sie interessant sind und welche automatisierten Aktionen durchgeführt werden sollen, wenn sich für ein Ereignis eine Übereinstimmung mit einer Regel ergibt.

> **Important**
>
> Ereignisbenachrichtigungen stellenAWS -Services typischerweise in wenigen Sekunden bereit CloudWatch , manchmal kann dies aber auch eine Minute oder länger dauern.

Wenn eine Datei in einen Container hochgeladen oder aus einem Container entfernt wird, werden im CloudWatch Dienst zwei Ereignisse nacheinander ausgelöst:

1. the section called “Object State Change Ereignis”
2. the section called “Container State Change Ereignis”

Informationen zum Abonnieren dieser Ereignisse finden Sie bei Amazon CloudWatch.

Die folgenden Aktionen können beispielsweise automatisch ausgelöst werden:

- Aufrufen einer AWS Lambda-Funktion
- Aufrufen eines Amazon EC2 Run Command
- Weiterleiten des Ereignisses an Amazon Kinesis Data Streams
- Aktivieren eines AWS Step Functions-Zustandsautomaten
- Benachrichtigen eines Amazon SNS-Themas oder einerAWS SMS Warteschlange

Beispiele für die Verwendung von CloudWatch Ereignissen mit AWS Elemental MediaStore :

- Aktivieren einer Lambda-Funktion bei jeder Erstellung eines Containers
- Benachrichtigen eines Amazon SNS-Themas, wenn ein Objekt gelöscht wird

Weitere Informationen finden Sie im Amazon CloudWatch Events-Benutzerhandbuch.

Themen

- Ereignis zur Änderung des MediaStore Objektstatus in AWS Elemental
- Ereignis zur Änderung des Status des MediaStore Containers in AWS Elemental

Ereignis zur Änderung des MediaStore Objektstatus in AWS Elemental

Dieses Ereignis wird veröffentlicht, wenn sich der Status eines Objekts geändert hat (wenn das Objekt hochgeladen oder gelöscht wurde).

> **Note**
>
> Objekte, die aufgrund einer transienten Datenregel ablaufen, geben kein CloudWatch Ereignis aus, wenn sie ablaufen.

Informationen zum Abonnieren dieser Veranstaltung finden Sie bei Amazon CloudWatch.

Objekt aktualisiert

```
{
  "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"
  }
}
```

Objekt entfernt

```
{
  "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"
  }
}
```

Ereignis zur Änderung des Status des MediaStore Containers in AWS Elemental

Dieses Ereignis wird veröffentlicht, wenn sich der Status eines Containers geändert hat (wenn ein Container hinzugefügt oder gelöscht wurde). Informationen zum Abonnieren dieser Veranstaltung finden Sie bei Amazon CloudWatch.

Container erstellt

```
{
  "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 entfernt

```
{
  "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"
  }
}
```

Überwachung von AWS Elemental MediaStore mit CloudWatch Amazon-Metriken

Sie können AWS Elemental MediaStore mithilfe von Berechtigungen CloudWatch überwachen. Dabei werden Rohdaten gesammelt und zu lesbaren Metriken verarbeitet. CloudWatchDie von gesammelten Statistiken werden 15 Monate gespeichert, damit Sie auf Verlaufsinformationen zugreifen können und einen besseren Überblick darüber erhalten, wie Ihre Webanwendung oder der Service ausgeführt wird. Sie können auch Alarme einrichten, die auf bestimmte Grenzwerte achten und Benachrichtigungen senden oder Aktivitäten auslösen, wenn diese Grenzwerte erreicht werden. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

Für AWS Elemental MediaStore möchten Sie vielleicht beobachten`BytesDownloaded` und sich selbst eine E-Mail senden, wenn diese Metrik einen bestimmten Schwellenwert erreicht.

So zeigen Sie Metriken mithilfe der CloudWatch -Konsole an:

Metriken werden zunächst nach dem Service-Namespace und anschließend nach den verschiedenen Dimensionskombinationen in den einzelnen Namespaces gruppiert.

1. Melden Sie sich bei der anAWS Management Console und öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie unter Alle Metriken den AWS/MediaStore Namespace aus.
4. Wählen Sie die Metrikdimension aus, um die Metriken anzuzeigen. Wählen Sie beispielsweise `Request metrics by container` aus, um Metriken für die verschiedenen Arten von Anforderungen anzuzeigen, die an den Container gesendet wurden.

So zeigen Sie Metriken mit der a AWS CLI

- Geben Sie als Eingabeaufforderung den folgenden Befehl ein:

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

AWS Elemental MediaStore Elemental-Metriken

In der folgenden Tabelle sind die Metriken aufgeführt, an die AWS Elemental MediaStore sendet CloudWatch.

> **Note**
>
> Um Metriken anzuzeigen, müssen Sie dem Container eine Kennzahlrichtlinie hinzufügen, um das Senden von Metriken an Amazon zu ermöglichen MediaStore CloudWatch.

Metrik	Beschreibung
`RequestCount`	Die Gesamtzahl der an einen MediaStore-Container gestellten HTTP-Anforderungen, getrennt durch den Vorgangstyp (`Put`, `Get`, `Delete`, `Describe`, `List`). Einheiten: Anzahl Gültige Dimensionen: • Container-Name • Objektgruppenname • Typ der Anforderung

Metrik	Beschreibung
	Gültige Statistiken: Summe
`4xxErrorCount`	Die Anzahl der HTTP-Anfragen MediaStore führte zu einem 4xx-Fehler. Einheiten: Anzahl Gültige Dimensionen: • Container-Name • Objektgruppenname • Typ der Anforderung Gültige Statistiken: Summe
`5xxErrorCount`	Die Anzahl der HTTP-Anfragen MediaStore führte zu einem 5xx-Fehler. Einheiten: Anzahl Gültige Dimensionen: • Container-Name • Objektgruppenname • Typ der Anforderung Gültige Statistiken: Summe

Metrik	Beschreibung
`BytesUploaded`	Die Anzahl der Bytes, die für Anforderungen an einen MediaStore -Container hochgeladen wurden, wobei die Anforderung einen Text enthält. Einheiten: Byte Gültige Dimensionen: • Container-Name • Objektgruppenname Gültige Statistiken: Durchschnitt (Byte pro Anforderung), Summe (Byte pro Zeitraum), Stichprobenanzahl, Min (entspricht P0,0), Max (entspricht p100), beliebiges Perzentil zwischen p0,0 und p99,9
`BytesDownloaded`	Anzahl der heruntergeladenen Bytes für Anforderungen an einen MediaStore -Container, wobei die Antwort einen Text enthält. Einheiten: Byte Gültige Dimensionen: • Container-Name • Objektgruppenname Gültige Statistiken: Durchschnitt (Byte pro Anforderung), Summe (Byte pro Zeitraum), Stichprobenanzahl, Min (entspricht P0,0), Max (entspricht p100), beliebiges Perzentil zwischen p0,0 und p99,9

Metrik	Beschreibung
`TotalTime`	Die Anzahl der Millisekunden (ms), die die Anforderung aus Perspektive des Servers unterwegs war. Dieser Wert wird ab der Zeit gemessen, zu der Ihre Anforderung MediaStore empfangen wurde, bis zu der Zeit, zu der das letzte Byte der Antwort gesendet wurde. Dieser Wert wird von der Perspektive des Servers aus gemessen, da Messungen aus der Perspektive des Clients von der Netzwerklatenz beeinträchtigt sind. Einheiten: Millisekunden Gültige Dimensionen: • Container-Name • Objektgruppenname • Typ der Anforderung Gültige Statistiken: Durchschnitt, Min (entspricht P0,0), Max (entspricht p100), beliebiges Perzentil zwischen p0,0 und p100
`TurnaroundTime`	Die Anzahl der Millisekunden, die MediaStore Ihre Anfrage verarbeitet hat. Dieser Wert wird ab der Zeit gemessen, zu der das letzte Byte Ihrer Anforderung MediaStore empfangen wurde, bis zu der Zeit, zu der das erste Byte der Antwort gesendet wurde. Einheiten: Millisekunden Gültige Dimensionen: • Container-Name • Objektgruppenname • Typ der Anforderung Gültige Statistiken: Durchschnitt, Min (entspricht P0,0), Max (entspricht p100), beliebiges Perzentil zwischen p0,0 und p100

Metrik	Beschreibung
`ThrottleCount`	Die Anzahl der HTTP-Anforderungen MediaStore , die an diese gestellt wurden, wurde gedrosselt. Einheiten: Anzahl Gültige Dimensionen: • Container-Name • Objektgruppenname • Typ der Anforderung Gültige Statistiken: Summe

Taggen von AWS Elemental Elemental-Ressourcen MediaStore

Ein Tag ist eine benutzerdefinierte Attributbezeichnung, die Sie oder AWS einer AWS-Ressource zuweisen. Jedes Tag besteht aus zwei Teilen:

- einem Tag-Schlüssel (z. B. `CostCenter`, `Environment` oder `Project`). Bei Tag-Schlüsseln wird zwischen Groß- und Kleinschreibung unterschieden.
- einem optionalen Feld, dem sogenannten Tag-Wert (z. B. `111122223333` oder `Production`). Ein nicht angegebener Tag-Wert entspricht einer leeren Zeichenfolge. Wie bei Tag-Schlüsseln wird auch bei Tag-Werten zwischen Groß- und Kleinschreibung unterschieden.

Tags sind für folgende Aktivitäten nützlich:

- Identifizieren und Organisieren Ihrer AWS-Ressourcen. Viele AWS-Services unterstützen das Markieren mit Tags (kurz: Tagging). So können Ressourcen aus verschiedenen Services denselben Tag zuweisen, um anzugeben, dass die Ressourcen verbunden sind. Sie könnten beispielsweise einem AWS Elemental MediaStore *`Elemental-Container`* dasselbe Tag zuweisen, das Sie einer AWS Elemental MediaLive Eingabe zuweisen.
- Überwachen Ihrer AWS-Kosten. Sie aktivieren diese Tags auf dem AWS Billing and Cost Management-Dashboard. AWS verwendet die Tags zur Kategorisierung Ihrer Kosten und zur

Bereitstellung eines monatlichen Kostenzuordnungsberichts für Sie. Weitere Informationen finden Sie unter Verwendung von Kostenzuordnungs-Tags im AWS Billing-Benutzerhandbuch.

In den folgenden Abschnitten finden Sie weitere Informationen zu Tags für AWS Elemental MediaStore.

Unterstützte Ressourcen in AWS Elemental MediaStore

Die folgenden Ressourcen in AWS Elemental MediaStore unterstützen Tagging:

- *Container*

Weitere Informationen zum Hinzufügen und Verwalten von Tags finden Sie unter Verwalten von Tags.

AWS Elemental unterstützt die tagbasierte Zugriffskontrollfunktion von AWS Identity and Access Management (IAM) MediaStore nicht.

Konventionen für die Tag-Benennung und -Verwendung

Die folgenden grundlegenden Benennungs- und Verwendungskonventionen gelten für die Verwendung von Tags mit AWS Elemental MediaStore Elemental-Ressourcen:

- Jede Ressource kann maximal 50 Tags haben.
- Jeder Tag muss für jede Ressource eindeutig sein. Jeder Tag kann nur einen Wert haben.
- Die maximale Länge des Tag-Schlüssels beträgt 128 Unicode-Zeichen in UTF-8.
- Die maximale Länge des Tag-Wertes beträgt 256 Unicode-Zeichen in UTF-8.
- Erlaubte Zeichen sind Buchstaben, Ziffern und Leerzeichen, die in UTF-8 darstellbar sind, sowie die folgenden Zeichen: . : + = @ _ / - (Bindestrich). Für Amazon EC2-Ressourcen können beliebige Zeichen verwendet werden.
- Bei Tag-Schlüsseln und -Werten muss die Groß- und Kleinschreibung beachtet werden. Eine bewährte Methode besteht darin, sich für eine einheitliche Schreibweise der Tag-Benennungen zu entscheiden und diese Strategie für alle Ressourcentypen umzusetzen. Entscheiden Sie sich beispielsweise für `Costcenter`, `costcenter` oder `CostCenter` und verwenden Sie diese Konvention für alle Tags. Vermeiden Sie die Verwendung von ähnlichen Tags mit uneinheitlicher Fallunterscheidung.

- Das Präfix `aws:` darf nicht für Tags verwendet werden; es ist für die AWS-Verwendung reserviert. Sie können keine Tag-Schlüssel oder -Werte mit diesem Präfix bearbeiten oder löschen. Tags mit diesem Präfix werden nicht auf Ihre Tags pro Ressourcenkontingent angerechnet.

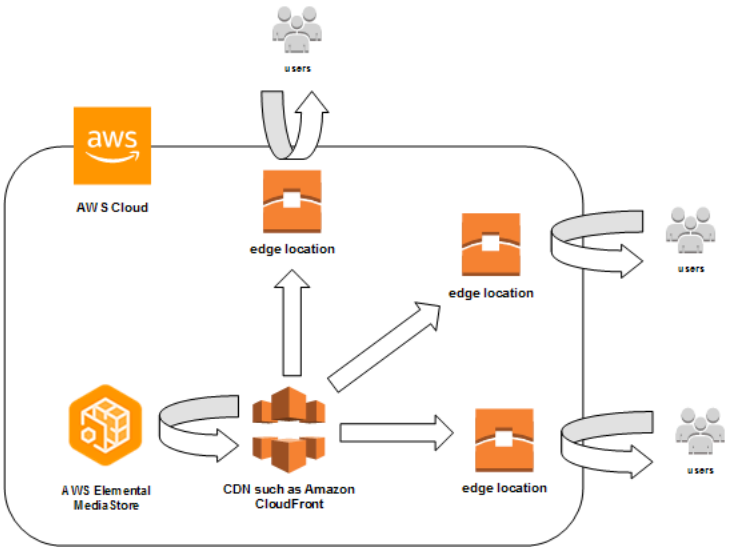
Verwalten von Tags

Tags bestehen aus den Eigenschaften `Key` und `Value` für eine Ressource. Sie können die AWS CLI oder die MediaStore API verwenden, um die Werte für diese Eigenschaften hinzuzufügen, zu bearbeiten oder zu löschen. Informationen zur Arbeit mit Tags finden Sie in den folgenden Abschnitten der AWS Elemental MediaStore API-Referenz:

- CreateContainer
- ListTagsForResource
- Ressourcen
- TagResource
- UntagResource

Arbeiten mit Netzwerken zur Bereitstellung von Inhalten (Content Delivery Networks, CDNs)

Sie können ein Content Delivery Network (CDN) wie Amazon verwenden, CloudFront um die Inhalte bereitzustellen, die Sie in AWS Elemental speichern MediaStore. Ein CDN ist eine weltweit verteilte Gruppe von Servern, die Inhalte wie Videos zwischenspeichern. Wenn ein Benutzer Ihren Inhalt anfordert, leitet das CDN die Anfrage an den Edge-Standort weiter, der die niedrigste Latenz bietet. Wenn Ihr Inhalt bereits an diesem Edge-Standort zwischengespeichert ist, stellt das CDN ihn unmittelbar bereit. Wenn sich Ihre Inhalte derzeit nicht an diesem Edge-Standort befinden, ruft das CDN sie von Ihrem Ursprung (z. B. Ihrem MediaStore Container) ab und verteilt sie an den Benutzer.



Themen

- Ermöglichen Sie Amazon CloudFront den Zugriff auf Ihren AWS Elemental MediaStore Elemental-Container
- Interaktion MediaStore von AWS Elemental mit HTTP-Caches

Ermöglichen Sie Amazon CloudFront den Zugriff auf Ihren AWS Elemental MediaStore Elemental-Container

Sie können Amazon verwenden CloudFront , um die Inhalte bereitzustellen, die Sie in einem Container in AWS Elemental speichern MediaStore. Sie können dafür eine der folgenden Möglichkeiten auswählen:

- Origin Access Control (OAC) verwenden- (Empfohlen) Verwenden Sie diese Option, wenn Sie die OAC-Funktion vonAWS-Region unterstützen CloudFront.
- Shared Secrets verwenden- Verwenden Sie diese Option, wenn Sie die OAC-Funktion vonAWS-Region nicht unterstützen CloudFront.

Origin Access Control (OAC) verwenden

Sie können die Origin Access Control (OAC) -Funktion von Amazon verwenden, CloudFront um die MediaStore Ursprünge von AWS Elemental mit verbesserter Sicherheit zu sichern. Sie können AWSSignature Version 4 (SigV4) für CloudFront Quellanfragen aktivieren MediaStore und festlegen, wann und ob die Anfragen signiert CloudFront werden sollen. Sie können CloudFront über die Konsole, APIs, SDK oder CLI auf die OAC-Funktion zugreifen, und für deren Nutzung fallen keine zusätzlichen Gebühren an.

Weitere Informationen zur Verwendung der OAC-Funktion mit MediaStore finden Sie unter Restricting access to a MediaStore origin im Amazon CloudFront Developer Guide.

Shared Secrets verwenden

Wenn Sie die OAC-Funktion von AmazonAWS-Region nicht unterstützen CloudFront, können Sie Ihrem AWS Elemental MediaStore Elemental-Container eine Richtlinie anhängen, die Lesezugriff oder mehr gewährt CloudFront.

> **Note**
>
> Wir empfehlen, die OAC-Funktion zu verwenden, wenn Sie sieAWS-Region unterstützen. Die folgenden Verfahren erfordern die Konfiguration von MediaStore und CloudFront mit Shared Secrets, um den Zugriff auf MediaStore Container einzuschränken. Um den besten Sicherheitspraktiken zu folgen, erfordert diese manuelle Konfiguration eine regelmäßige Rotation der Geheimnisse. Mit OAC on MediaStore Origins können Sie anweisen, Anfragen mit SigV4 CloudFront zu signieren und sie MediaStore zum Signaturabgleich an sie weiterzuleiten, sodass keine Secrets verwendet und rotiert werden müssen. Dadurch wird sichergestellt, dass Anfragen automatisch überprüft werden, bevor Medieninhalte bereitgestellt werden, wodurch die Bereitstellung von Medieninhalten CloudFront einfacher MediaStore und sicherer wird.

Um den CloudFront Zugriff auf Ihren Container zu ermöglichen (Konsole)

1. Öffnen Sie die MediaStore Konsole unter https://console.aws.amazon.com/mediastore/.
2. Wählen Sie auf der Seite Containers (Container) den Containernamen aus.

   Die Seite mit den Containerdetails wird angezeigt.
3. Fügen Sie im Abschnitt Container-Richtlinie eine Richtlinie bei, die Amazon Lesezugriff oder mehr gewährt CloudFront.

   Example

   Die folgende Beispielrichtlinie, die der Beispielrichtlinie für Public Read Access über HTTPS ähnelt, erfüllt diese Anforderungen, da sie allen Personen, die Anfragen über HTTPS an Ihre Domain senden, erlaubt`GetObject` und`DescribeObject` Befehle erteilt. Darüber hinaus schützt die folgende Beispielrichtlinie Ihren Workflow besser, da sie den CloudFront Zugriff auf MediaStore Objekte nur zulässt, wenn die Anforderung über eine HTTPS-Verbindung erfolgt und den richtigen Referer-Header enthält.

```
{
  "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. Weisen Sie im Abschnitt Container CORS policy (Container CORS-Richtlinie) eine Richtlinie zu, die die entsprechende Zugriffsebene gewährt.

   > **Note**
   >
   > Eine CORS-Richtlinie ist nur dann erforderlich, wenn Sie Zugriff auf einen browserbasierten Player gewähren wollen.

5. Notieren Sie dabei die folgenden Details:

   - Den Datenendpunkt, der Ihrem Container zugeordnet ist. Sie finden diese Informationen im Abschnitt Info auf der Seite Containers (Container). CloudFrontIn wird der Datenendpunkt als Quelldomänenname bezeichnet.
   - Die Ordnerstruktur im Container, in dem die Objekte gespeichert werden. CloudFrontIn wird dies als Quellpfad bezeichnet. Beachten Sie, dass diese Einstellung optional ist. Weitere Informationen zu Ausgangspfaden finden Sie im Amazon CloudFront Developer Guide.

6. Erstellen Sie in CloudFront eine Distribution, die für die Bereitstellung von Inhalten von AWS Elemental konfiguriert ist MediaStore. Sie benötigen dazu die Informationen, die Sie im vorigen Schritt aufgezeichnet haben.

Nachdem Sie die Richtlinie an Ihre MediaStore Container angehängt haben, müssen Sie CloudFront sie so konfigurieren, dass nur HTTPS-Verbindungen für Quellanfragen verwendet werden, und außerdem einen benutzerdefinierten Header mit dem richtigen geheimen Wert hinzufügen.

So konfigurieren Sie den CloudFront Zugriff auf Ihren Container über eine HTTPS-Verbindung mit einem geheimen Wert für den Referer-Header (Konsole)

1. Öffnen Sie die CloudFront -Konsole.
2. Wähle auf der Origins-Seite deinen MediaStore Ursprung aus.
3. Wählen Sie Edit (Bearbeiten) aus.
4. Wählen Sie HTTPS nur für das Protokoll.
5. Wählen Sie im Abschnitt Benutzerdefinierte Kopfzeile hinzufügen die Option Kopfzeile hinzufügen aus.
6. Wählen Sie als Namen Referer aus. Verwenden Sie für den Wert dieselbe <secretValue>Zeichenfolge, die Sie in Ihrer Container-Richtlinie verwendet haben.
7. Wählen Sie Speichern und lassen Sie die Änderungen wirksam werden.

Interaktion MediaStore von AWS Elemental mit HTTP-Caches

AWS Elemental MediaStore speichert Objekte, sodass sie von Content Delivery Networks (CDNs) wie Amazon korrekt und effizient zwischengespeichert werden können CloudFront. Wenn ein Endbenutzer oder ein CDN ein Objekt von abruft MediaStore, gibt der Dienst HTTP-Header zurück, die das Caching-Verhalten des Objekts beeinflussen. (Die Standards für das Caching-Verhalten bei HTTP 1.1 finden Sie in RFC2616, Abschnitt 13.) Diese Header sind:

- **`ETag`** (nicht anpassbar) – Der Header für das Entitäts-Tag ist eine eindeutige Kennung für die Antwort, die MediaStore sendet. Standardkonforme CDNs und Webbrowser verwenden dieses Tag als Schlüssel, mit dem das Objekt zwischengespeichert wird. MediaStore generiert automatisch eine`ETag` für jedes Objekt, wenn es hochgeladen wird. Sie können die Details eines Objekts anzeigen, um seinen ETag-Wert zu ermitteln.
- **`Last-Modified`**(nicht anpassbar) — Der Wert dieses Headers gibt das Datum und die Uhrzeit der Änderung des Objekts an. MediaStore generiert diesen Wert automatisch, wenn das Objekt hochgeladen wird.
- **`Cache-Control`** (anpassbar) – Der Wert dieses Headers steuert, wie lange ein Objekt zwischengespeichert werden soll, bevor das CDN überprüft, ob es geändert wurde. Sie können diesen Header auf einen beliebigen Wert setzen, wenn Sie ein Objekt mithilfe der CLI oder API in einen MediaStore Container hochladen. Eine Übersicht über alle gültigen Werte finden Sie in der HTTP/1.1-Dokumentation. Wenn Sie diesen Wert nicht festlegen, wenn Sie ein Objekt hochladen, MediaStore wird dieser Header nicht zurückgegeben, wenn das Objekt abgerufen wird.

  Der Cache-Control-Header wird meist dazu verwendet, um die Zwischenspeicherdauer für ein Objekt anzugeben. Angenommen, Ihre Videomanifestdatei wird häufig von einem Encoder überschrieben. Sie können den Wert für `max-age` auf 10 setzen, damit das Objekt nur 10 Sekunden lang zwischengespeichert wird. Ein weiteres Beispiel ist ein gespeichertes Videosegment, das nie überschrieben wird. Sie können den Wert für `max-age` auf 31 536 000 festlegen, damit das Objekt etwa 1 Jahr lang zwischengespeichert wird.

Bedingte Anforderungen

Bedingte Anfragen an MediaStore

MediaStore reagiert identisch auf bedingte Anfragen (unter Verwendung von Anforderungsheadern wie`If-Modified-Since` und`If-None-Match`, wie in RFC7232 beschrieben) und bedingungslose

Anfragen. Das bedeutet, dass der Dienst beim MediaStore Empfang einer gültigen`GetObject` Anfrage das Objekt immer zurückgibt, auch wenn der Client das Objekt bereits besitzt.

Bedingte Anforderungen an CDNs

CDNs, die Inhalte im Namen von bereitstellen, MediaStore können bedingte Anfragen bearbeiten`304 Not Modified`, indem sie zurückkehren, wie in RFC7232 Abschnitt 4.1 beschrieben. Es muss also nicht der gesamte Objektinhalt übertragen werden, weil der Anforderer bereits ein Objekt hat, das zur bedingten Anforderung passt.

CDNs (und andere Caches, die mit HTTP/1.1 kompatibel sind) treffen diese Entscheidungen basierend auf den Headern `Cache-Control` und `ETag`, die von den Ursprungsservern weitergeleitet werden. Um zu kontrollieren, wie oft CDNs MediaStore Originalserver nach Updates für wiederholt abgerufene Objekte abfragen, legen Sie die`Cache-Control` Header für diese Objekte fest, wenn Sie sie hochladen MediaStore.

Kontingente in AWS Elemental MediaStore

Die Service Quotas Quotas-Konsole bietet Informationen zu AWS Elemental MediaStore Quotas. Neben der Anzeige der Standardkontingente können Sie die Servicekontingentkonsole verwenden, um Kontingenterhöhungen für einstellbare Kontingente anzufordern.

In der folgenden Tabelle werden Kontingente, die früher als Grenzwerte bezeichnet wurden, in AWS Elemental beschrieben MediaStore. Kontingente sind die maximale Anzahl von Serviceressourcen oder -vorgängen für Ihr AWS-Konto.

> **Note**
>
> Um einzelnen Containern in Ihrem Konto Kontingente zuzuweisen, wenden Sie sich an den AWS-Support oder Ihren Kundenbetreuer. Diese Option kann Ihnen helfen, die Limits auf Kontoebene auf Ihre Container aufzuteilen, um zu verhindern, dass ein Container Ihr gesamtes Kontingent verbraucht.

Ressource oder Operation	Standardkontingent	Kommentare
Container	100	Die maximale Anzahl der Container, die Sie in diesem Konto erstellen können.
Ordnerebenen	10	Die maximale Anzahl der Ordnerebenen, die Sie in einem Container erstellen können. Sie können eine beliebige Anzahl von Ordnern erstellen, solange sie nicht mehr als 10 Ebenen innerhalb eines Containers verschachtelt sind.
Ordner	Unbegrenzt	Sie können eine beliebige Anzahl von Ordnern erstellen, solange sie nicht mehr als 10 Ebenen innerhalb eines Containers verschachtelt sind.
Objektgröße	25 MB	Die maximale Dateigröße eines einzelnen Objekts.

Ressource oder Operation	Standardkontingent	Kommentare
Objekte	Unbegrenzt	Sie können beliebig viele Objekte in einen Ordner oder Container in Ihrem Konto hochladen.
Rate of DeleteObject API requests (Rate der API-Anforderungen)	100	Die maximale Anzahl von Operationsanforderungen, die Sie pro Sekunde machen können. Darüber hinausgehende Anfragen werden gedrosselt. Sie können eine Kontingenterhöhung beantragen.
Rate of DescribeObject API requests (Rate der API-Anforderungen)	1.000	Die maximale Anzahl von Operationsanforderungen, die Sie pro Sekunde machen können. Darüber hinausgehende Anfragen werden gedrosselt. Sie können eine Kontingenterhöhung beantragen.
Rate der GetObject API-Anforderungen für Standard-Upload-Verfügbarkeit	1.000	Die maximale Anzahl von Operationsanforderungen, die Sie pro Sekunde machen können. Darüber hinausgehende Anfragen werden gedrosselt. Sie können eine Kontingenterhöhung beantragen.
Rate der GetObject API-Anforderungen für Streaming-Upload-Verfügbarkeit	25	Die maximale Anzahl von Operationsanforderungen, die Sie pro Sekunde machen können. Darüber hinausgehende Anfragen werden gedrosselt. Sie können eine Kontingenterhöhung beantragen.

Ressource oder Operation	Standardkontingent	Kommentare
Rate of ListItems API requests (Rate der API-Anfor derungen)	5	Die maximale Anzahl von Operationsanforder ungen, die Sie pro Sekunde machen können. Darüber hinausgehende Anfragen werden gedrosselt. Sie können eine Kontingenterhöhung beantragen.
Rate der PutObject API-Anforderungen für aufgeteilte Übertragungscodier ung (auch bekannt als Streaming-Upload-Verfügbark eit)	10	Die maximale Anzahl von Operationsanforder ungen, die Sie pro Sekunde machen können. Darüber hinausgehende Anfragen werden gedrosselt. Sie können eine Kontingenterhöhung beantrage n. Geben Sie in der Anforderung die angeforderte TPS und die durchschnittliche Objektgröße an.
Rate der PutObject API-Anforderungen für Standard-Upload-Verfügbarke it	100	Die maximale Anzahl von Operationsanforder ungen, die Sie pro Sekunde machen können. Darüber hinausgehende Anfragen werden gedrosselt. Sie können eine Kontingenterhöhung beantrage n. Geben Sie in der Anforderung die angeforderte TPS und die durchschnittliche Objektgröße an.
Regeln in einer Metrikrichtlinie	10	Die maximale Anzahl von Regeln, die Sie in eine Metrikrichtlinie aufnehmen können.
Regeln in einer Objektlebenszyklus -Richtlinie	10	Die maximale Anzahl der Regeln, die in einer Objektlebenszyklus-Richtlinie enthalten sein können.

Informationen zu AWS MediaStore Elemental

Die folgende Tabelle enthält verwandte Ressourcen, die bei der Arbeit mit AWS Elemental nützlich sind MediaStore.

- Kurse und Workshops — Links zu rollenbasierten und speziellen Kursen sowie Übungen im Selbststudium zur Verbesserung IhrerAWS -Kompetenzen und für praktische Erfahrung.
- AWSDeveloper Center — Entdecken Sie Tutorials, laden Sie Tools herunter und erfahren Sie mehr über Veranstaltungen fürAWS -Entwickler.
- AWS-Entwickler-Tools — Links zu Entwickler-Tools, SDKs, IDE-Toolkits und Befehlszeilen-Tools für die Entwicklung und Verwaltung vonAWS -Anwendungen.
- Ressourcenzentrum für die ersten Schritte — Hier erfahren SieAWS-Konto, wie Sie einrichten, derAWS Community beitreten und Ihre erste Anwendung starten.
- Praktische Tutorials — step-by-step Schritt-für-Schritt-für-Schritt-Anleitungen zum Starten Ihrer ersten Anwendung aufAWS.
- AWS-Whitepaper — Links zu einer umfangreichen Liste technischerAWS -Whitepaper zu Themen wie Architektur, Sicherheit und Wirtschaftlichkeit. Diese Whitepaper wurden vonAWS -Lösungsarchitekten und anderen technischen Experten verfasst.
- AWS Support-Center – Hub für die Erstellung und Verwaltung Ihrer AWS Support-Fälle. Stellt darüber hinaus Links zu weiteren nützlichen Ressourcen bereit, beispielsweise Foren, häufig gestellten technischen Fragen, Status der Service-Integrität und AWS Trusted Advisor.
- AWS Support— Primäre Website für Informationen zuAWS Support one-on-one, einem reaktionsschnellen Support-Channel, der Sie bei der Erstellung und Ausführung von Anwendungen in der Cloud unterstützt.
- Kontakt – Zentraler Kontaktpunkt für Fragen zu AWS-Abrechnung, Konten, Ereignissen Missbrauch und anderen Problemen.
- Nutzungsbedingungen für die AWS-Website – Detaillierte Informationen zu unseren Copyright- und Markenbestimmungen, Ihrem Konto, den Lizenzen und anderen Themen.

Dokumentverlauf für das Benutzerhandbuch

Die folgende Tabelle beschreibt die Dokumentation zu dieser Version von AWS Elemental MediaStore. Um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten, können Sie einen RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Verbesserung der Origin Access Control (OAC)	wurden wurden Informationen zur Verwendung von OAC mit AWS Elemental hinzugefügt MediaStore.	17. April 2023
Aktualisierungen der Kontingente	Der Quotenwert und die Beschreibung für wurden korrigiert`Rules in a Metric Policy`.	25. Oktober 2022
ExpiresAt Feld	Zugriffsprotokolle enthalten jetzt ein`ExpiresAt` Feld, das das Ablaufdatum und die Gültigkeitsdauer des Objekts angibt, basierend auf vorübergehenden Datenrege ln in der Lebenszyklusrichtlinie des Containers.	16. Juli 2020
Regeln für den Lebenszyk lusübergang	Sie können nun eine Lebenszyklus-Übergangsregel zu Ihrer Objektlebenszyklus -Richtlinie hinzufügen, die festlegt, dass Objekte nach Erreichen eines bestimmten Alters in die Speicherklasse für seltenen Zugriff verschoben werden.	20. April 2020

Leerer Behälter	Sie können nun alle Objekte innerhalb eines Containers gleichzeitig löschen.	7. April 2020
Support für CloudWatch Amazon-Metriken	Sie können eine Kennzahlr ichtlinie festlegen, an welche Metriken MediaStore gesendet CloudWatch werden.	30. März 2020
Platzhalter in den Regeln zum Löschen von Objekten	In einer Richtlinie zum Objektlebenszyklus können Sie nun einen Platzhalter in einer Regel zum Löschen von Objekten verwenden. Damit können Sie Dateien anhand ihres Dateinamens oder ihrer Erweiterung angeben, die der Service nach einer bestimmte n Anzahl von Tagen löschen soll.	20. Dezember 2019
Richtlinien für den Lebenszyk lus von Objekten	Sie können Ihrer Objektleb enszyklus-Richtlinie jetzt eine Regel hinzufügen, die einen Ablauf nach Alter in Sekunden anzeigt.	13. September 2019

AWS CloudFormation--Support	Sie können nun eine AWS CloudFormation-Vorlage verwenden, um automatisch einen Container zu erstellen . Die AWS CloudFormation-Vorlage verwaltet Daten für fünf API-Aktionen: Erstellung eines Containers, Einstellung der Zugriffsprotokollierung, Aktualisierung der Standard-Containerrichtlinie, Hinzufügen einer CORS-Richtlinie (Cross-Origin Resource Sharing) und Hinzufügen einer Objektlebenszyklusrichtlinie.	17. Mai 2019
Kontingente für Streaming-Upload-Verfügbarkeit	Bei Objekten mit Streaming-Upload-Verfügbarkeit (gestückelte Übertragung von Objekten) darf die `PutObject` -Operation 10 TPS und die `GetObject` -Operation 25 TPS nicht überschreiten.	8. April 2019
Übertragung von Objekten in Blöcken	Zusätzliche Unterstützung für das Aufteilen der Übertragung von Objekten. Mit dieser Funktion können Sie angeben, dass ein Objekt zum Herunterladen verfügbar ist, bevor das Objekt vollständig hochgeladen ist.	5. April 2019

Zugriffsprotokollierung	AWS Elemental unterstützt MediaStore jetzt Zugriffsprotokollierung, bei der es sich um detaillierte Aufzeichnungen über die Anfragen handelt, die an Objekte in einem Container gestellt wurden.	25. Februar 2019
Richtlinien für den Lebenszyklus von Objekten	Zusätzliche Unterstützung für Objektlebenszyklus-Richtlinien, die das Ablaufdatum von Objekten innerhalb des aktuellen Containers regeln.	12. Dezember 2018
Erhöhtes Kontingent für Objektgröße	Das Kontingent für die Größe eines Objekts beträgt nun 25 MB.	10. Oktober 2018
Erhöhtes Kontingent für Objektgröße	Das Kontingent für die Größe eines Objekts beträgt nun 20 MB.	6. September 2018
AWS CloudTrail-Integration	Der CloudTrail Integrationsinhalt wurde aktualisiert, um ihn an die jüngsten Änderungen am CloudTrail Dienst anzupassen.	12. Juli 2018
CDN-Kooperation	Es wurden Informationen zur Verwendung von AWS Elemental MediaStore mit einem Content Delivery Network (CDN) wie Amazon hinzugefügt CloudFront.	14. April 2018

CORS-Konfigurationen	AWS Elemental unterstützt MediaStore jetzt Cross-Origin Resource Sharing (CORS) bestimmt für Client-Webanwendungen, die in einer Domain geladen sind, eine Möglichkeit zur Interaktion mit Ressourcen in einer anderen Domain.	7. Februar 2018
Neuer Service und neues Service und neues Service	Dies ist die erste Version des Dienstes zur Erstellung und Speicherung von Videos, AWS Elemental MediaStore, und des AWS Elemental MediaStore User Guide.	27. November 2017

Note

- DieAWS Mediendienste sind nicht für die Verwendung mit Anwendungen oder in Situationen konzipiert oder vorgesehen, in denen eine ausfallsichere Leistung erforderlich ist, wie z. B. lebenssichere Betriebsabläufe, Navigations- oder Kommunikationssysteme, Flugsicherungs- oder Lebenserhaltungssysteme, in denen die Nichtverfügbarkeit, Unterbrechung oder Ausfall der Dienste zu Tod, Personen-, Sach- oder Umweltschäden führen kann.

AWS-Glossar

Die neueste AWS-Terminologie finden Sie im AWS-Glossar in der AWS-Glossar-Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.