aws

Guide de l’utilisateur

AWS Elemental MediaStore

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Elemental MediaStore: Guide de l’utilisateur

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et la présentation commerciale d’Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n’est pas d’Amazon, d’une manière susceptible de créer une confusion parmi les clients, ou d’une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d’Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

..... vi
Qu'est-ce que c'est MediaStore ? 1
Concepts et terminologie 1
Services connexes 3
Accès MediaStore 3
Tarification 4
Régions et points de terminaison 4
Configuration d'AWS Elemental MediaStore 6
Inscrivez-vous pour un Compte AWS 6
Création d'un utilisateur doté d'un accès administratif 7
Premiers pas 9
Étape 1 : accéder à AWS Elemental MediaStore 9
Étape 2 : Créer un conteneur 9
Étape 3 : Charger un objet 10
Étape 4 : Accéder à un objet 11
Conteneurs 12
Règles relatives aux noms de conteneur 12
Création d'un conteneur 12
Affichage des détails du conteneur 14
Affichage d'une liste des conteneurs 15
Suppression d'un conteneur 16
Stratégies 17
Stratégies de conteneur 17
Affichage d'une stratégie de conteneur 18
Modification d'une stratégie de conteneur 19
Exemples de stratégies de conteneur 20
Stratégies CORS 28
Scénarios d'utilisation 28
Ajout d'une stratégie CORS 29
Affichage d'une stratégie CORS 30
Modification d'une stratégie CORS 31
Suppression d'une stratégie CORS 32
Résolution des problèmes 33
Exemples de stratégies CORS 34

Stratégies de cycle de vie des objets ... 36
Composants d'une stratégie de cycle de vie des objets ... 36
Ajout d'une stratégie de cycle de vie des objets ... 43
Affichage d'une stratégie de cycle de vie des objets ... 45
Modification d'une stratégie de cycle de vie des objets ... 46
Suppression d'une stratégie de cycle de vie des objets ... 47
Exemples de stratégie de cycle de vie des objets ... 48
Stratégies de métriques ... 52
Ajout d'une stratégie de métriques ... 53
Affichage d'une stratégie de métriques ... 54
Modification d'une stratégie de métriques ... 54
Exemples de stratégies de métriques ... 54
Dossiers ... 58
Règles des noms de dossier ... 59
Création d'un dossier ... 59
Suppression d'un dossier ... 59
Objets ... 61
Chargement d'un objet ... 61
Affichage d'une liste ... 63
Affichage des détails de l'objet ... 66
Téléchargement d'un objet ... 67
Suppression d'objets ... 68
Suppression d'un objet ... 68
Vidage d'un conteneur ... 69
Sécurité ... 71
Protection des données ... 72
Chiffrement des données ... 73
Gestion de l'identité et des accès ... 73
Public ciblé ... 74
Authentification par des identités ... 74
Gestion de l'accès à l'aide de politiques ... 76
Comment AWS Elemental MediaStore fonctionne avec IAM ... 77
Exemples de stratégies basées sur l'identité ... 84
Résolution de problème ... 87
Journalisation et surveillance ... 89
CloudWatch Alarmes Amazon ... 89

AWS CloudTrail journaux 89
AWS Trusted Advisor 90
Validation de conformité 90
Résilience 90
Sécurité de l'infrastructure 91
Prévention du cas de figure de l'adjoint désorienté entre services 91
Surveillance et balisage 94
Journalisation des appels d'API avec CloudTrail 95
MediaStoreInformations dans CloudTrail 96
Exemple : entrées de fichier journal 97
Surveillance avec CloudWatch 98
CloudWatch Journaux 99
CloudWatch Événements 109
CloudWatch métriques 113
Identification 118
Ressources prises en charge dans AWS Elemental MediaStore 119
Conventions de dénomination et d'utilisation des balises 119
Gestion des tags 120
Travailler avec CDNs 121
Autorisation de CloudFront à accéder à votre conteneur 121
Utilisation du contrôle d'accès d'origine (OAC) 122
Utilisation de secrets partagés 122
Interaction de MediaStoreavec les caches HTTP 124
Demandes conditionnelles 125
Travailler avec AWS SDKs 126
Exemples de code 128
Principes de base 128
Actions 129
Quotas 152
Informations connexes 155
Historique de la documentation 156
AWS Glossaire 161

Avis de fin de support : le 13 novembre 2025, le support d'AWS MediaStore Elemental AWS sera interrompu. Après le 13 novembre 2025, vous ne pourrez plus accéder à la MediaStore console ni aux MediaStore ressources. Pour plus d'informations, consultez ce billet de blog.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.

Qu'est-ce qu'AWS Elemental ? MediaStore

AWS Elemental MediaStore est un service de création et de stockage vidéo qui offre les hautes performances et la cohérence immédiate requises pour la création en direct. Vous pouvez ainsi gérer les ressources vidéo sous forme d'objets dans des conteneurs afin de créer des flux de travail multimédia fiables basés sur le cloud. MediaStore

Pour utiliser le service, vous chargez vos objets à partir d'une source, telle qu'un encodeur ou un flux de données, vers un conteneur que vous créez dans MediaStore.

MediaStore est un excellent choix pour stocker des fichiers vidéo fragmentés lorsque vous avez besoin d'une forte cohérence, d'une faible latence de lecture et d'écriture, et de la capacité de gérer de gros volumes de demandes simultanées. Si vous ne diffusez pas de vidéos en direct, pensez plutôt à utiliser Amazon Simple Storage Service (Amazon S3).

Rubriques

- MediaStore Concepts et terminologie AWS Elemental
- Services connexes
- Accès à AWS Elemental MediaStore
- Tarification d'AWS Elemental MediaStore
- Régions et points de terminaison pour AWS Elemental MediaStore

MediaStore Concepts et terminologie AWS Elemental

ARN

Un Amazon Resource Name.

Corps de texte

Les données à charger dans un objet.

Plage (octet)

Un sous-ensemble de données d'objet à attribuer. Pour plus d'informations, consultez plage à partir de la spécification HTTP.

Conteneur

Un espace de noms qui contient des objets. Un conteneur dispose d'un point de terminaison que vous pouvez utiliser pour l'écriture et la récupération d'objets ainsi que l'attachement de stratégies d'accès.

Point de terminaison

Point d'entrée du MediaStore service, indiqué sous la forme d'une URL racine HTTPS.

ETag

Une balise d'entité, qui est un hachage des données d'objet.

Dossier

Une division d'un conteneur. Un dossier peut contenir des objets et d'autres dossiers.

Élément

Un terme utilisé pour faire référence à des objets et des dossiers.

Objet

Un actif, similaire à un objet Amazon S3. Les objets sont les entités fondamentales stockées dans MediaStore. Le service accepte tous les types de fichiers.

Service de montage

MediaStore est considéré comme un service d'origine car c'est le point de distribution pour la diffusion du contenu multimédia.

Chemin

Un identifiant unique pour un objet ou un dossier, qui indique son emplacement dans le conteneur.

Partie

Un sous-ensemble de données (fragment) d'un objet.

Politique

Une stratégie IAM.

Ressource

Entité dans AWS que vous pouvez utiliser. Chaque ressource AWS se voit attribuer un Amazon Resource Name (ARN) qui tient lieu d'identifiant unique. Voici MediaStore la ressource et son format ARN :

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

Services connexes

- Amazon CloudFront est un service de réseau mondial de diffusion de contenu (CDN) qui fournit des données et des vidéos en toute sécurité à vos spectateurs. Utilisez CloudFront pour diffuser du contenu avec les meilleures performances possibles. Pour plus d'informations, consultez le manuel Amazon CloudFront Developer Guide.
- CloudFormationest un service qui vous aide à modéliser et à configurer vos AWS ressources. Vous créez un modèle qui décrit toutes les AWS ressources que vous souhaitez (comme les MediaStore conteneurs), et vous vous CloudFormation occupez du provisionnement et de la configuration de ces ressources pour vous. Vous n'avez pas besoin de créer et de configurer AWS des ressources individuellement et de déterminer ce qui dépend de quoi ; il CloudFormation gère tout cela. Pour plus d'informations, consultez le Guide de l'utilisateur AWS CloudFormation.
- AWS CloudTrailest un service qui vous permet de surveiller les appels passés à l' CloudTrail API de votre compte, y compris les appels passés par l'AWS Management Console et d'autres services. AWS CLI Pour plus d'informations, consultez le Guide de l'utilisateur AWS CloudTrail.
- Amazon CloudWatch est un service de surveillance des ressources du AWS cloud et des applications que vous utilisez AWS. Utilisez CloudWatch les événements pour suivre l'évolution de l'état des conteneurs et des objets dans MediaStore. Pour plus d'informations, consultez la CloudWatch documentation Amazon.
- Gestion des identités et des accès AWS (IAM) est un service Web qui vous permet de contrôler en toute sécurité l'accès aux AWS ressources pour vos utilisateurs. Utilisez IAM pour contrôler qui peut utiliser vos AWS ressources (authentification) et quelles ressources les utilisateurs peuvent utiliser et de quelle manière (autorisation). Pour de plus amples informations, veuillez consulter Configuration d'AWS Elemental MediaStore.
- Amazon Simple Storage Service (Amazon S3) est un système de stockage d'objets conçu pour stocker et récupérer n'importe quel volume de données, où que vous soyez. Pour plus d'informations, consultez la documentation Amazon S3.

Accès à AWS Elemental MediaStore

Vous pouvez y accéder MediaStore en utilisant l'une des méthodes suivantes :

- AWS Management Console : les procédures décrites dans ce guide expliquent comment utiliser l'AWS Management Console pour effectuer des tâches pour MediaStore. Pour y accéder MediaStore via la console :

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

- AWS Command Line Interface— Pour plus d'informations, consultez le guide de AWS Command Line Interface l'utilisateur. Pour accéder à l' MediaStore aide du point de terminaison de la CLI :

```
aws mediastore
```

- MediaStore API — Si vous utilisez un langage de programmation pour lequel aucun SDK n'est disponible, consultez la référence des AWS Elemental MediaStore API pour obtenir des informations sur les actions d'API et sur la manière de faire des demandes d'API. Pour accéder à l' MediaStore aide du point de terminaison de l'API REST :

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

- AWS SDKs — Si vous utilisez un langage de programmation pour lequel AWS fournit un SDK, vous pouvez utiliser un SDK pour y accéder. MediaStore SDKs simplifiez l'authentification, intégrez facilement votre environnement de développement et facilitez l'accès aux MediaStore commandes. Pour plus d'informations, consultez Outils pour Amazon Web Services.
- Outils AWS pour Windows PowerShell : pour plus d'informations, consultez le guide de Outils AWS pour PowerShell l'utilisateur.

Tarification d'AWS Elemental MediaStore

Comme pour les autres AWS produits, il n'existe aucun contrat ni engagement minimum d'utilisation MediaStore. Vous êtes facturé selon des frais d'intégration par Go lorsque le contenu pénètre dans le service et selon des frais mensuels par Go pour le contenu que vous stockez dans le service. Pour plus d'informations, consultez la section Tarification d'AWS Elemental MediaStore .

Régions et points de terminaison pour AWS Elemental MediaStore

Pour réduire la latence des données dans vos applications, MediaStore propose un point de terminaison régional pour effectuer votre demande :

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

Pour consulter la liste complète des régions AWS disponibles, consultez la section MediaStore Points de terminaison et quotas AWS Elemental dans le manuel de référence général AWS.
MediaStore

Configuration d'AWS Elemental MediaStore

Cette section vous guide à travers les étapes requises pour configurer les utilisateurs afin qu'ils accèdent à AWS Elemental MediaStore. Pour obtenir des informations générales et supplémentaires sur la gestion des identités et des accès pour MediaStore, voirIdentity and Access Management pour AWS Elemental MediaStore.

Pour commencer à utiliser AWS Elemental MediaStore, procédez comme suit.

Rubriques

- Inscrivez-vous pour un Compte AWS
- Création d'un utilisateur doté d'un accès administratif

Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez l'https://portal.aws.amazon.com/billing/inscription.
2. Suivez les instructions en ligne.

   Une partie de la procédure d'inscription consiste à recevoir un appel téléphonique ou un message texte et à saisir un code de vérification sur le clavier du téléphone.

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l'utilisateur racine pour effectuer les tâches nécessitant un accès utilisateur racine.

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte en accédant à https://aws.amazon.com/et en choisissant Mon compte.

Création d'un utilisateur doté d'un accès administratif

Une fois que vous vous êtes inscrit à un utilisateur administratif Compte AWS, que vous Utilisateur racine d'un compte AWS l'avez sécurisé AWS IAM Identity Center, que vous l'avez activé et que vous en avez créé un, afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en Console de gestion AWStant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

   Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez Connexion en tant qu'utilisateur racine dans le Guide de l'utilisateur Connexion à AWS .
2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur racine.

   Pour obtenir des instructions, consultez la section Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root (console) dans le guide de l'utilisateur IAM.

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez Activation d' AWS IAM Identity Center dans le Guide de l'utilisateur AWS IAM Identity Center .
2. Dans IAM Identity Center, octroyez un accès administratif à un utilisateur.

   Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center dans le Guide de AWS IAM Identity Center l'utilisateur.

Connexion en tant qu'utilisateur doté d'un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l'URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l'utilisateur IAM Identity Center.

Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAM Identity Center, consultez la section Connexion au portail AWS d'accès dans le guide de l'Connexion à AWS utilisateur.

Attribution d’un accès à d’autres utilisateurs

1. Dans IAM Identity Center, créez un ensemble d’autorisations qui respecte la bonne pratique consistant à appliquer les autorisations de moindre privilège.

   Pour obtenir des instructions, consultez Création d’un ensemble d’autorisations dans le Guide de l’utilisateur AWS IAM Identity Center .
2. Attribuez des utilisateurs à un groupe, puis attribuez un accès par authentification unique au groupe.

   Pour obtenir des instructions, consultez Ajout de groupes dans le Guide de l’utilisateur AWS IAM Identity Center .

Commencer à utiliser AWS Elemental MediaStore

Ce didacticiel de mise en route explique comment utiliser AWS Elemental MediaStore pour créer un conteneur et télécharger un objet.

Rubriques

- Étape 1 : accéder à AWS Elemental MediaStore
- Étape 2 : Créer un conteneur
- Étape 3 : Charger un objet
- Étape 4 : Accéder à un objet

Étape 1 : accéder à AWS Elemental MediaStore

Une fois que vous avez configuré votre compte AWS et créé des utilisateurs et des rôles, vous vous connectez à la console pour AWS Elemental MediaStore.

Pour accéder à AWS Elemental MediaStore

- Connectez-vous à la MediaStore console Console de gestion AWS et ouvrez-la à l'adresse https://console.aws.amazon.com/mediastore/.

> **Note**
>
> Vous pouvez vous connecter à l'aide de n'importe quelle information d'identification IAM créée pour ce compte. Pour plus d'informations sur la création d'informations d'identification IAM, consultez Configuration d'AWS Elemental MediaStore.

Étape 2 : Créer un conteneur

Vous utilisez des conteneurs dans AWS Elemental MediaStore pour stocker vos dossiers et vos objets. Vous pouvez utiliser des conteneurs pour regrouper des objets connexes, comme lorsque vous utilisez un répertoire pour regrouper des fichiers dans un système de fichiers. Vous n'êtes pas facturé lorsque vous créez des conteneurs, mais uniquement lorsque vous chargez un objet dans un conteneur.

Pour créer un conteneur

1. Sur la page Containers (Conteneurs), choisissez Create container (Créer un conteneur).
2. Pour Container name (Nom du conteneur), saisissez un nom pour votre conteneur. Pour de plus amples informations, veuillez consulter Règles relatives aux noms de conteneur.
3. Choisissez Créer un conteneur. AWS Elemental MediaStore ajoute le nouveau conteneur à une liste de conteneurs. Initialement, le statut du conteneur est Creating (Création en cours), puis il passe à Active (Actif).

Étape 3 : Charger un objet

Vous pouvez charger des objets (jusqu'à 25 Mo chacun) dans un conteneur ou dans un dossier du conteneur. Pour charger un objet dans un dossier, vous spécifiez le chemin d'accès au dossier. Si le dossier existe déjà, AWS Elemental y MediaStore stocke l'objet. Si le dossier n'existe pas, le service le crée, puis stocke l'objet dans le dossier.

> **Note**
>
> Les noms de fichiers d'objet doivent être composés uniquement de lettres, de chiffres, de points (.), de traits de soulignement (_), de tildes (~), et de traits d'union (-).

Pour charger un objet

1. Sur la page Containers (Conteneurs), choisissez le nom du conteneur que vous venez de créer. La page des détails du conteneur s'affiche.
2. Choisissez Upload object (Charger un objet).
3. Pour Target path (Chemin d'accès cible), saisissez un chemin pour les dossiers. Par exemple, `premium/canada`. Si l'un des dossiers du chemin n'existe pas encore, AWS Elemental les MediaStore crée automatiquement.
4. Pour Object (Objet), choisissez Browse (Parcourir).
5. Naviguez jusqu'au dossier approprié et choisissez un objet à charger.
6. Choisissez Open (Ouvrir), puis Upload (Charger).

Étape 4 : Accéder à un objet

Vous pouvez télécharger vos objets vers un point de terminaison spécifié.

1. Sur la page Containers (Conteneurs), choisissez le nom du conteneur pour lequel vous souhaitez télécharger l'objet.
2. Si l'objet que vous souhaitez télécharger se trouve dans un sous-dossier, choisissez les noms de dossier jusqu'à ce que vous voyiez l'objet.
3. Choisissez le nom de l'objet.
4. Sur la page des détails de l'objet, choisissez Download (Télécharger).

Conteneurs dans AWS Elemental MediaStore

Vous utilisez des conteneurs MediaStore pour stocker vos dossiers et vos objets. Des objets connexes peuvent être regroupés dans des conteneurs de la même façon que vous utilisez un répertoire pour regrouper des fichiers dans un système de fichiers. Vous n'êtes pas facturé lorsque vous créez des conteneurs, mais uniquement lorsque vous chargez un objet dans un conteneur. Pour plus d'informations sur les frais, consultez la section Tarification d'AWS Elemental MediaStore.

Rubriques

- Règles relatives aux noms de conteneur
- Création d'un conteneur
- Affichage des détails d'un conteneur
- Affichage d'une liste des conteneurs
- Suppression d'un conteneur

Règles relatives aux noms de conteneur

Lorsque vous choisissez un nom pour votre conteneur, n'oubliez pas les points suivants :

- Le nom doit être unique dans le compte actuel pour la région AWS actuelle.
- Le nom peut contenir des lettres majuscules, des lettres minuscules, des chiffres et des traits de soulignements (_).
- Il doit comporter entre 1 et 255 caractères.
- Les noms sont sensibles à la casse. Par exemple, un conteneur peut être nommé `myContainer` et un dossier peut être nommé `mycontainer` car ces noms sont uniques.
- Un conteneur ne peut pas être renommé après avoir été créé.

Création d'un conteneur

Vous pouvez créer jusqu'à 100 conteneurs pour chaque compte AWS. Vous pouvez créer autant de dossiers que vous le souhaitez, à condition qu'ils ne soient pas imbriqués sur plus de 10 niveaux au sein d'un conteneur. En outre, vous pouvez charger autant d'objets que vous le souhaitez dans chaque conteneur.

> **Tip**
>
> Vous pouvez également créer un conteneur automatiquement à l'aide d'un CloudFormation modèle. Le modèle CloudFormation gère les données pour les cinq actions d'API : création d'un conteneur, définition de la journalisation des accès, mise à jour de la stratégie de conteneur par défaut, ajout d'une stratégie de partage des ressources cross-origin (CORS), et en ajoutant une stratégie de cycle de vie des objets. Pour plus d'informations, consultez le Guide de l'utilisateur AWS CloudFormation.

Pour créer un conteneur (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez Create container (Créer un conteneur).
3. Pour Container name (Nom du conteneur), saisissez un nom pour votre conteneur. Pour de plus amples informations, veuillez consulter Règles relatives aux noms de conteneur.
4. Choisissez Créer un conteneur. AWS Elemental MediaStore ajoute le nouveau conteneur à une liste de conteneurs. Initialement, le statut du conteneur est Creating (Création en cours), puis il passe à Active (Actif).

Pour créer un conteneur (AWS CLI)

- Dans le AWS CLI, utilisez la `create-container` commande :

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

L'exemple suivant illustre la valeur de retour :

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

```
}
```

Affichage des détails d'un conteneur

Les détails d'un conteneur incluent la stratégie de conteneur, le point de terminaison, l'ARN et l'heure de création.

Pour afficher la page des détails d'un conteneur (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur.

   La page des détails du conteneur s'affiche. Cette page est divisée en deux sections :

   - La section Objects (Objets), qui répertorie les objets et les dossiers du conteneur.
   - La section Container policy (Stratégie de conteneur), qui affiche la stratégie basée sur les ressources qui est associée à ce conteneur. Pour plus d'informations sur les stratégies de ressources, consultez Stratégies de conteneur.

Pour afficher la page des détails d'un conteneur (AWS CLI)

- Dans le AWS CLI, utilisez la `describe-container` commande :

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

  L'exemple suivant illustre la valeur de retour :

  ```
  {
      "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"
      }
  ```

```
}
```

Affichage d'une liste des conteneurs

Vous pouvez afficher une liste de tous les conteneurs associés à votre compte.

Pour afficher une liste des conteneurs (console)

- Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.

  La page Containers (Conteneurs) s'affiche, répertoriant tous les conteneurs associés à votre compte.

Pour afficher une liste des conteneurs (AWS CLI)

- Dans le AWS CLI, utilisez la `list-containers` commande.

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

  L'exemple suivant illustre la valeur de retour :

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

Suppression d'un conteneur

Vous pouvez supprimer un conteneur uniquement s'il ne possède aucun objet.

Pour supprimer un conteneur (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez l'option à gauche du nom du conteneur.
3. Sélectionnez Delete (Supprimer).

Pour supprimer un conteneur (AWS CLI)

- Dans le AWS CLI, utilisez la `delete-container` commande :

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

  Cette commande ne renvoie aucune valeur.

Politiques dans AWS Elemental MediaStore

Vous pouvez appliquer une ou plusieurs des politiques suivantes à votre conteneur AWS Elemental MediaStore :

- Politique de conteneur : définit les droits d'accès à tous les dossiers et objets du conteneur. MediaStore définit une politique par défaut qui permet aux utilisateurs d'effectuer toutes les MediaStore opérations sur le conteneur. Cette stratégie spécifie que toutes les opérations doivent être effectuées via HTTPS. Après avoir créé un conteneur, vous pouvez modifier la stratégie de conteneur.
- Politique de partage des ressources entre origines (CORS) : permet aux applications Web clientes d'un domaine d'interagir avec les ressources d'un autre domaine. MediaStore ne définit pas de politique CORS par défaut.
- Politique relative aux métriques : permet MediaStore d'envoyer des métriques à Amazon CloudWatch. MediaStore ne définit pas de politique métrique par défaut.
- Politique de cycle de vie des objets : contrôle la durée pendant laquelle les objets restent dans un MediaStore conteneur. MediaStorene définit pas de politique de cycle de vie des objets par défaut.

Politiques relatives aux conteneurs dans AWS Elemental MediaStore

Chaque conteneur inclut une stratégie basée sur les ressources qui régit les droits d'accès à tous les dossiers et objets dans ce conteneur. La politique par défaut, qui est automatiquement attachée à tous les nouveaux conteneurs, permet d'accéder à toutes les MediaStore opérations AWS Elemental sur le conteneur. Elle spécifie que cet accès exige HTTPS pour les opérations. Une fois que vous avez créé un conteneur, vous pouvez modifier la stratégie attachée à ce conteneur.

Vous pouvez également utiliser une stratégie de cycle de vie des objets qui régit la date d'expiration des objets dans un conteneur. Une fois que les objets ont atteint l'ancienneté maximale que vous spécifiez, le service les supprime du conteneur.

Rubriques

- Affichage d'une stratégie de conteneur
- Modification d'une stratégie de conteneur
- Exemples de stratégies de conteneur

Affichage d'une stratégie de conteneur

Vous pouvez utiliser la console ou le AWS CLI pour afficher la politique basée sur les ressources d'un conteneur.

Pour afficher une stratégie de conteneur (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur.

   La page des détails du conteneur s'affiche. La stratégie s'affiche dans la section Container policy (Stratégie de conteneur).

Pour afficher une stratégie de conteneur (AWS CLI)

- Dans le AWS CLI, utilisez la `get-container-policy` commande :

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

L'exemple suivant illustre la valeur de retour :

```
{
  "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"
```

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

Modification d'une stratégie de conteneur

Vous pouvez modifier les autorisations de la stratégie de conteneur par défaut, ou créer une nouvelle stratégie qui remplace celle par défaut. Cinq minutes peuvent s'écouler avant que la nouvelle stratégie prenne effet.

Pour modifier une stratégie de conteneur (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur.
3. Choisissez Modifier la politique. Pour obtenir des exemples illustrant la définition des différentes autorisations, consultez the section called "Exemples de stratégies de conteneur".
4. Effectuez les modifications appropriées, puis choisissez Save (Enregistrer).

Pour modifier une stratégie de conteneur (AWS CLI)

1. Créez un fichier qui définit la stratégie de conteneur :

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadOverHttps",
      "Effect": "Allow",
      "Action": ["mediastore:GetObject", "mediastore:DescribeObject"],
      "Principal": "*",
      "Resource": "arn:aws:mediastore:us-
west-2:111122223333:container/ExampleLiveDemo/*",
      "Condition": {
        "Bool": {
```

```
            "aws:SecureTransport": "true"
          }
        }
      }
    ]
}
```

2. Dans le AWS CLI, utilisez la `put-container-policy` commande :

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

Cette commande ne renvoie aucune valeur.

Exemples de stratégies de conteneur

Les exemples suivants illustrent les stratégies qui sont créées pour différents groupes d'utilisateurs.

Rubriques

- Exemple de stratégie de conteneur : Par défaut
- Exemple de stratégie de conteneur : Accès public en lecture sur HTTPS
- Exemple de stratégie de conteneur : Accès public en lecture sur HTTP ou HTTPS
- Exemple de stratégie de conteneur : Accès en lecture entre comptes -compatible HTTP
- Exemple de stratégie de conteneur : Accès en lecture entre comptes sur HTTPS
- Exemple de stratégie de conteneur : Accès en lecture entre comptes à un rôle
- Exemple de stratégie de conteneur : Accès complet en lecture entre comptes à un rôle
- Exemple de stratégie de conteneur : Accès restreint à des adresses IP spécifiques

Exemple de stratégie de conteneur : Par défaut

Lorsque vous créez un conteneur, AWS Elemental joint MediaStore automatiquement la politique basée sur les ressources suivante :

JSON

```
{
```

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

La stratégie est intégrée au service, vous n'avez donc pas à la créer. Toutefois, vous pouvez modifier la politique du conteneur si les autorisations de la stratégie par défaut ne correspondent pas aux autorisations que vous souhaitez utiliser pour le conteneur.

La stratégie par défaut attribuée à tous les nouveaux conteneurs accorde l'accès à toutes les opérations MediaStore sur le conteneur. Elle spécifie que cet accès exige HTTPS pour les opérations.

Exemple de stratégie de conteneur : Accès public en lecture sur HTTPS

Cet exemple de stratégie permet aux utilisateurs de récupérer un objet via une demande HTTPS. Il permet à quiconque d'accéder en lecture via une SSL/TLS connexion sécurisée : utilisateurs authentifiés et utilisateurs anonymes (utilisateurs non connectés). L'instruction est nommée `PublicReadOverHttps`. Elle accorde l'accès aux opérations `GetObject` et `DescribeObject` sur n'importe quel objet (tel que spécifié par * à la fin du chemin d'accès à la ressource). Elle spécifie que cet accès exige HTTPS pour les opérations :

JSON

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

Exemple de stratégie de conteneur : Accès public en lecture sur HTTP ou HTTPS

Cet exemple de stratégie accorde l'accès aux opérations `GetObject` et `DescribeObject` sur n'importe quel objet (tel que spécifié par * à la fin du chemin d'accès à la ressource). Il accorde l'accès en lecture à tout le monde, notamment tous les utilisateurs authentifiés et les utilisateurs anonymes (les utilisateurs qui ne sont pas connectés) :

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadOverHttpOrHttps",
            "Effect": "Allow",
            "Action": [
```

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

Exemple de stratégie de conteneur : Accès en lecture entre comptes -compatible HTTP

Cette stratégie permet aux utilisateurs de récupérer un objet via une demande HTTP. Elle accorde cet accès aux utilisateurs authentifiés avec l'accès entre comptes. Il n'est pas nécessaire que l'objet soit hébergé sur un serveur doté d'un SSL/TLS certificat :

JSON

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

```
                    "aws:SecureTransport": "true"
                }
            }
        }
    ]
}
```

Exemple de stratégie de conteneur : Accès en lecture entre comptes sur HTTPS

Cet exemple de politique autorise l'accès aux `DescribeObject` opérations `GetObject` et sur n'importe quel objet (tel que spécifié par le * à la fin du chemin de ressource) appartenant à l'utilisateur root de l'utilisateur spécifié<other acct number>. Elle spécifie que cet accès exige HTTPS pour les opérations :

JSON

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

Exemple de stratégie de conteneur : Accès en lecture entre comptes à un rôle

Cet exemple de stratégie accorde l'accès aux opérations `GetObject` et `DescribeObject` sur n'importe quel objet (tel que spécifié par * à la fin du chemin d'accès à la ressource) qui appartient au <propriétaire du numéro de compte>. Elle accorde cet accès à n'importe quel utilisateur de l'<autre numéro de compte> si ce compte a endossé le rôle spécifié dans le <nom du rôle> :

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

Exemple de stratégie de conteneur : Accès complet en lecture entre comptes à un rôle

Cet exemple de stratégie autorise l'accès entre comptes pour mettre à jour n'importe quel objet dans le compte, tant que l'utilisateur est connecté sur HTTP. Il accorde également l'accès entre comptes pour supprimer, télécharger et décrire les objets sur HTTP ou HTTPS à un compte qui a endossé le rôle spécifié :

- La première instruction est nommée `CrossAccountRolePostOverHttps`. Elle autorise l'accès à l'opération `PutObject` sur n'importe quel objet et accorde cet accès à n'importe quel utilisateur du compte spécifié si ce compte a endossé le rôle spécifié dans le <nom de rôle>. Elle spécifie que cet accès exige HTTPS pour l'opération (cette condition doit toujours être incluse lors de l'accord d'un accès à `PutObject`).

  En d'autres termes, toute personne habilitée disposant d'un accès entre comptes peut accéder à `PutObject`, mais uniquement sur HTTPS.
- La deuxième instruction est nommée `CrossAccountFullAccessExceptPost`. Elle accorde l'accès à toutes les opérations à l'exception de `PutObject` sur n'importe quel objet. Elle accorde

cet accès à n'importe quel utilisateur du compte spécifié si ce compte a endossé le rôle spécifié dans le <nom de rôle>. Cet accès n'exige pas HTTPS pour les opérations.

En d'autres termes, chaque compte disposant d'un accès entre comptes peut accéder à `DeleteObject`, `GetObject`, etc. (à l'exception de `PutObject`), et sur HTTP ou HTTPS.

Si vous n'excluez pas `PutObject` de la deuxième instruction, cette dernière ne sera pas valide (car si vous incluez `PutObject` vous devez explicitement définir HTTPS en tant que condition).

JSON

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

Exemple de stratégie de conteneur : Accès restreint à des adresses IP spécifiques

Cet exemple de politique permet d'accéder à toutes les MediaStore opérations AWS Elemental sur les objets du conteneur spécifié. Toutefois, la demande doit provenir de la plage d'adresses IP indiquée dans la condition.

La condition contenue dans cette déclaration identifie la plage 198.51.100.* d'adresses IP autorisées du protocole Internet version 4 (IPv4), à une exception près : 198.51.100.188.

Le bloc `Condition` utilise les conditions `IpAddress` et `NotIpAddress` et la clé de condition `aws:SourceIp`, qui est une clé de condition à l'échelle d'AWS. Les `aws:sourceIp` IPv4 valeurs utilisent la notation CIDR standard. Pour plus d'informations, consultez la section Opérateurs de condition d'adresse IP dans le guide de l'utilisateur IAM.

JSON

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

```
}
```

Politiques de partage de ressources entre origines (CORS) dans AWS Elemental MediaStore

Le partage des ressources cross-origin (CORS) définit un moyen pour les applications Web clientes chargées dans un domaine particulier d'interagir avec les ressources d'un autre domaine. Grâce à la prise en charge du CORS dans AWS MediaStore Elemental, vous pouvez créer des applications Web riches côté client MediaStore avec et autoriser de manière sélective l'accès à vos ressources entre différentes origines. MediaStore

> **Note**
>
> Si vous utilisez Amazon CloudFront pour distribuer du contenu à partir d'un conteneur doté d'une politique CORS, veillez à configurer la distribution pour AWS MediaStore Elemental (y compris l'étape de modification du comportement du cache pour configurer CORS).

Cette section fournit une présentation du CORS. Les sous-rubriques décrivent comment activer CORS à l'aide de la MediaStore console AWS Elemental ou par programmation à l'aide de l'API REST et d'AWS. MediaStore SDKs

Rubriques

- Scénarios d'utilisation CORS
- Ajout d'une stratégie CORS à un conteneur
- Affichage d'une stratégie CORS
- Modification d'une stratégie CORS
- Suppression d'une stratégie CORS
- Dépannage des problèmes liés à la stratégie CORS
- Exemples de stratégies CORS

Scénarios d'utilisation CORS

Les exemples de scénarios suivants utilisent le CORS :

- Scénario 1 : Supposons que vous distribuiez une vidéo en direct dans un MediaStore conteneur AWS Elemental nommé. LiveVideo Vos utilisateurs chargent le point de terminaison du manifeste vidéo `http://livevideo.mediastore.ap-southeast-2.amazonaws.com` à partir d'une origine spécifique telle que `www.example.com`. Vous souhaitez utiliser un lecteur JavaScript vidéo pour accéder aux vidéos provenant de ce conteneur via des demandes `GET` et `PUT` des requêtes non authentifiées. Un navigateur bloque généralement l'autorisation JavaScript de ces demandes, mais vous pouvez définir une politique CORS sur votre conteneur pour autoriser explicitement ces demandes depuis`www.example.com`.
- Scénario 2 : Supposons que vous souhaitiez héberger le même flux en direct que dans le scénario 1 à partir de votre MediaStore conteneur, mais que vous souhaitiez autoriser les demandes provenant de n'importe quelle origine. Vous pouvez configurer une stratégie CORS pour autoriser les origines de caractère générique (*), afin que les demandes de n'importe quelle origine puissent accéder à la vidéo.

Ajout d'une stratégie CORS à un conteneur

Cette section explique comment ajouter une configuration de partage de ressources entre origines (CORS) à un conteneur AWS MediaStore Elemental. La spécification CORS permet aux applications Web clientes chargées dans un domaine particulier d'interagir avec les ressources d'un autre domaine.

Pour configurer votre conteneur afin d'autoriser les demandes cross-origin, vous ajoutez une stratégie CORS au conteneur. Une stratégie CORS définit les règles identifiant les origines auxquelles vous autorisez l'accès à votre conteneur, les opérations (méthodes HTTP) prises en charge pour chaque origine, et d'autres informations propres aux opérations.

Lorsque vous ajoutez une stratégie CORS au conteneur, les stratégies de conteneur (qui régissent les droits d'accès au conteneur) continuent de s'appliquer.

Pour ajouter une stratégie CORS (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur pour lequel vous souhaitez créer une stratégie CORS.

   La page des détails du conteneur s'affiche.
3. Dans la section Container CORS policy (Stratégie CORS du conteneur), choisissez Create CORS policy (Créer une stratégie CORS).

4. Insérez la stratégie au format JSON, puis choisissez Save (Enregistrer).

Pour ajouter une stratégie CORS (AWS CLI)

1. Créez un fichier qui définit la stratégie CORS :

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

2. Dans le AWS CLI, utilisez la `put-cors-policy` commande.

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

Cette commande ne renvoie aucune valeur.

Affichage d'une stratégie CORS

Le partage des ressources cross-origin (CORS) définit un moyen pour les applications Web clientes chargées dans un domaine particulier d'interagir avec les ressources d'un autre domaine.

Pour afficher une stratégie CORS (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur pour lequel vous souhaitez afficher la stratégie CORS.

La page des détails du conteneur s'affiche, avec la stratégie CORS dans la section Container CORS policy (Stratégie CORS du conteneur).

Pour afficher une stratégie CORS (AWS CLI)

- Dans le AWS CLI, utilisez la `get-cors-policy` commande :

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

L'exemple suivant illustre la valeur de retour :

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

Modification d'une stratégie CORS

Le partage des ressources cross-origin (CORS) définit un moyen pour les applications Web clientes chargées dans un domaine particulier d'interagir avec les ressources d'un autre domaine.

Pour modifier une stratégie CORS (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.

2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur pour lequel vous souhaitez modifier la stratégie CORS.

   La page des détails du conteneur s'affiche.

3. Dans la section Container CORS policy (Stratégie CORS du conteneur), choisissez Edit CORS policy (Modifier une stratégie CORS).

4. Modifiez la stratégie, puis choisissez Save (Enregistrer).

Pour modifier une stratégie CORS (AWS CLI)

1. Créez un fichier qui définit la stratégie CORS mise à jour :

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

2. Dans le AWS CLI, utilisez la `put-cors-policy` commande.

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

   Cette commande ne renvoie aucune valeur.

Suppression d'une stratégie CORS

Le partage des ressources cross-origin (CORS) définit un moyen pour les applications Web clientes chargées dans un domaine particulier d'interagir avec les ressources d'un autre domaine. La

suppression de la stratégie CORS d'un conteneur supprime les autorisations pour les demandes cross-origin.

Pour supprimer une stratégie CORS (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur pour lequel vous souhaitez supprimer la stratégie CORS.

   La page des détails du conteneur s'affiche.
3. Dans la section Container CORS policy (Stratégie CORS du conteneur), choisissez Delete CORS policy (Supprimer une stratégie CORS).
4. Choisissez Continue (Continuer) pour confirmer, puis Save (Enregistrer).

Pour supprimer une stratégie CORS (AWS CLI)

- Dans le AWS CLI, utilisez la `delete-cors-policy` commande :

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

  Cette commande ne renvoie aucune valeur.

Dépannage des problèmes liés à la stratégie CORS

En cas de comportement inattendu lors de l'accès à un conteneur comportant une stratégie CORS, suivez les étapes ci-dessous pour résoudre le problème.

1. Vérifiez que la stratégie CORS est attachée au conteneur.

   Pour obtenir des instructions, consultez the section called “Affichage d'une stratégie CORS”.
2. Capturez la demande et la réponse complètes grâce à l'outil de votre choix (comme la console du développeur de votre navigateur). Vérifiez que la stratégie CORS attachée au conteneur inclut au moins une règle CORS correspondant aux données de votre demande, comme suit :

   a. Vérifiez que la demande possède un en-tête `Origin`.

Si l'en-tête est absent, AWS Elemental MediaStore ne traite pas la demande comme une demande d'origine croisée et ne renvoie pas les en-têtes de réponse CORS dans la réponse.

b. Vérifiez que l'en-tête `Origin` de votre demande correspond au moins à l'un des éléments `AllowedOrigins` de la règle `CORSRule` spécifique.

   La méthode, l'hôte et les valeurs de port de l'en-tête de demande `Origin` doivent correspondre à l'élément `AllowedOrigins` de la règle `CORSRule`. Par exemple, si vous configurez la règle `CORSRule` pour autoriser l'origine `http://www.example.com`, alors les deux origines `https://www.example.com` et `http://www.example.com:80` de votre demande ne correspondent pas à l'origine autorisée dans votre configuration.

c. Vérifiez que la méthode de votre demande (ou la méthode spécifiée dans la demande `Access-Control-Request-Method` en cas de demande en amont) correspond à l'un des éléments `AllowedMethods` de la même règle `CORSRule`.

d. Pour une demande en amont, si la demande inclut un en-tête `Access-Control-Request-Headers`, vérifiez que la règle `CORSRule` inclut les entrées `AllowedHeaders` pour chaque valeur dans l'en-tête `Access-Control-Request-Headers`.

Exemples de stratégies CORS

Les exemples suivants illustrent des stratégies de partage des ressources cross-origin (CORS).

Rubriques

- Exemple de stratégie CORS : Accès en lecture de n'importe quel domaine
- Exemple de stratégie CORS : Accès en lecture d'un domaine spécifique

Exemple de stratégie CORS : Accès en lecture de n'importe quel domaine

La politique suivante permet à une page Web de n'importe quel domaine de récupérer le contenu de votre conteneur AWS Elemental MediaStore . La demande inclut tous les en-têtes HTTP du domaine d'origine, et le service répond uniquement aux demandes HTTP GET et HTTP HEAD à partir du domaine d'origine. Les résultats sont mis en cache pendant 3 000 secondes avant qu'un nouvel ensemble de résultats soit diffusé.

```
[
```

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

Exemple de stratégie CORS : Accès en lecture d'un domaine spécifique

La politique suivante autorise une page Web `https://www.example.com` à récupérer le contenu de votre conteneur AWS Elemental MediaStore. La demande inclut tous les en-têtes HTTP de `https://www.example.com`, et le service répond uniquement aux demandes HTTP GET et HTTP HEAD à partir de `https://www.example.com`. Les résultats sont mis en cache pendant 3 000 secondes avant qu'un nouvel ensemble de résultats soit diffusé.

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

Politiques relatives au cycle de vie des objets dans AWS Elemental MediaStore

Pour chaque conteneur, vous pouvez créer une stratégie de cycle de vie des objets qui régit la durée pendant laquelle les objets doivent être stockés dans le conteneur. Lorsque les objets atteignent l'âge maximum que vous spécifiez, AWS Elemental MediaStore les supprime. Vous pouvez supprimer des objets dès que vous n'en avez plus besoin pour économiser sur les coûts de stockage.

Vous pouvez également spécifier qui MediaStore doit déplacer les objets vers la classe de stockage à accès peu fréquent (IA) une fois qu'ils ont atteint un certain âge. Les objets stockés dans la classe de stockage IA (accès peu fréquent) ont des taux de stockage et d'extraction différents de ceux stockés dans la classe de stockage standard. Pour plus d'informations, consultez MediaStore Pricing (Tarification CTlong).

Une stratégie de cycle de vie des objets contient des règles qui dictent la durée de vie des objets par sous-dossier. (Vous ne pouvez pas attribuer une stratégie de cycle de vie des objets à des objets individuels). Vous pouvez attacher une seule stratégie de cycle de vie des objets à un conteneur, mais vous pouvez ajouter jusqu'à 10 règles à chaque stratégie de cycle de vie des objets. Pour de plus amples informations, veuillez consulter Composants d'une stratégie de cycle de vie des objets.

Rubriques

- Composants d'une stratégie de cycle de vie des objets
- Ajout d'une stratégie de cycle de vie des objets à un conteneur
- Affichage d'une stratégie de cycle de vie des objets
- Modification d'une stratégie de cycle de vie des objets
- Suppression d'une stratégie de cycle de vie des objets
- Exemples de stratégie de cycle de vie des objets

Composants d'une stratégie de cycle de vie des objets

Les politiques relatives au cycle de vie des objets régissent la durée pendant laquelle les objets restent dans un conteneur AWS Elemental MediaStore . Chaque stratégie de cycle de vie des objets se compose d'une ou de plusieurs règles qui déterminent la durée de vie des objets. Une règle peut s'appliquer à un seul dossier, à plusieurs dossiers ou à l'ensemble du conteneur.

Vous pouvez attacher une stratégie de cycle de vie des objets à un conteneur et chaque stratégie de cycle de vie des objets peut contenir jusqu'à 10 règles. Vous ne pouvez pas attribuer une stratégie de cycle de vie des objets à un objet individuel.

Règles d'une stratégie de cycle de vie des objets

Vous pouvez créer trois types de règle :

- Données transitoires
- Suppression d'objet
- Transition du cycle de vie

Données transitoires

Une règle de données transitoires définit les objets pour qu'ils expirent en quelques secondes. Ce type de règle s'applique uniquement aux objets ajoutés au conteneur après l'entrée en vigueur de la stratégie. L'application de la nouvelle politique MediaStore au conteneur prend jusqu'à 20 minutes.

Voici un exemple de règle pour les données transitoires :

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

Les règles de données transitoires comportent trois parties :

- `path` : toujours défini sur `wildcard`. Vous utilisez cette partie pour définir les objets que vous souhaitez supprimer. Vous pouvez utiliser un ou plusieurs caractères génériques, représentés par un astérisque (*). Chaque caractère générique représente n'importe quelle combinaison de zéro caractère ou plus. Par exemple, `"path": [ {"wildcard": "Football/index*.m3u8"} ],` s'applique à tous les fichiers du dossier `Football` qui correspondent au modèle `index*.m3u8` (par exemple, index.m3u8, index1.m3us8 et index123456.m3u8). Vous pouvez inclure jusqu'à 10 chemins de dans une même règle.

- `seconds_since_create` : toujours défini sur `numeric`. Vous pouvez spécifier une valeur comprise entre 1 et 300 secondes. Vous pouvez également définir l'opérateur sur « supérieur à » (**>**) ou « supérieur ou égal à » (**>=**).
- `action` : toujours défini sur `EXPIRE`.

Pour les règles de données transitoires (les objets expirent en quelques secondes), il n'y a pas de décalage entre l'expiration d'un objet et la suppression de l'objet.

> **Note**
>
> Les objets qui sont soumis à une règle de données transitoires ne sont pas inclus dans une réponse `list-items`. En outre, les objets qui expirent en raison d'une règle de données transitoire n'émettent pas d' CloudWatch événement lorsqu'ils expirent.

Suppression d'objet

Une règle de suppression d'objet définit les objets pour qu'ils expirent en quelques jours. Ce type de règle s'applique à tous les objets du conteneur, même s'ils ont été ajoutés au conteneur avant la création de la stratégie. L'application de la nouvelle politique prend jusqu' MediaStore à 20 minutes, mais le retrait des objets du conteneur peut prendre jusqu'à 24 heures.

Voici un exemple de deux règles pour supprimer des objets :

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

```
            "action": "EXPIRE"
        }
```

Les règles de suppression d'objet comportent trois parties :

- `path` : Définissez sur `prefix` ou `wildcard`. Vous ne pouvez pas mélanger `prefix` et `wildcard` dans la même règle. Si vous souhaitez utiliser les deux, vous devez créer une règle pour `prefix` et une règle distincte pour `wildcard`, comme indiqué dans l'exemple ci-dessus.
  - `prefix` – Vous définissez le chemin de `prefix` si vous souhaitez supprimer tous les objets d'un dossier particulier. Si le paramètre est vide (`"path": [ { "prefix": "" } ],`), la cible correspond à tous les objets stockés n'importe où dans le conteneur actuel. Vous pouvez inclure jusqu'à 10 chemins de `prefix` dans une même règle.
  - `wildcard` – Vous définissez le chemin de `wildcard` si vous souhaitez supprimer des objets spécifiques en fonction du nom de fichier et/ou du type de fichier. Vous pouvez utiliser un ou plusieurs caractères génériques, représentés par un astérisque (*). Chaque caractère générique représente n'importe quelle combinaison de zéro caractère ou plus. Par exemple, `"path": [ {"wildcard": "Football/*.ts"} ],` s'applique à tous les fichiers du dossier `Football` qui correspondent au modèle `*.ts` (tels que nomfichier.ts, nomfichier1.ts et nomfichier123456.ts). Vous pouvez inclure jusqu'à 10 chemins de `wildcard` dans une même règle.
- `days_since_create` : toujours défini sur `numeric`. Vous pouvez spécifier une valeur comprise entre 1 et 36 500 jours. Vous pouvez également définir l'opérateur sur « supérieur à » (**>**) ou « supérieur ou égal à » (**>=**).
- `action` : toujours défini sur `EXPIRE`.

Pour les règles de suppression d'objet (les objets expirent en quelques jours), il peut y avoir un léger décalage entre l'expiration d'un objet et la suppression de l'objet. Toutefois, les modifications relatives à la facturation se produisent dès que l'objet expire. Par exemple, si une règle de cycle de vie spécifie 10 `days_since_create`, le compte n'est pas facturé pour l'objet une fois que celui-ci a 10 jours, même s'il n'est pas encore supprimé.

Transition du cycle de vie

Une règle de transition du cycle de vie définit les objets à déplacer vers la classe de stockage IA (accès peu fréquent) une fois qu'ils ont atteint un certain âge, mesuré en jours. Les objets stockés dans la classe de stockage IA (accès peu fréquent) ont des taux de stockage et d'extraction

différents de ceux stockés dans la classe de stockage standard. Pour plus d'informations, consultez MediaStore Pricing (Tarification CTlong).

Lorsqu'un objet a été déplacé vers la classe de stockage IA (accès peu fréquent), vous ne pouvez pas le déplacer vers la classe de stockage standard.

La règle de transition du cycle de vie s'applique à tous les objets du conteneur, même s'ils ont été ajoutés au conteneur avant la création de la stratégie. L'application de la nouvelle politique prend jusqu' MediaStore à 20 minutes, mais le retrait des objets du conteneur peut prendre jusqu'à 24 heures.

Vous trouverez ci-après un exemple de règle de transition du cycle de vie :

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

Les règles de transition du cycle de vie comportent trois parties :

- `path` : Définissez sur `prefix` ou `wildcard`. Vous ne pouvez pas mélanger `prefix` et `wildcard` dans la même règle. Si vous souhaitez utiliser les deux, vous devez créer une règle pour `prefix` et une règle distincte pour `wildcard`.
  - `prefix` - Vous définissez le chemin d'accès à `prefix` si vous souhaitez transférer tous les objets d'un dossier particulier vers la classe de stockage IA (accès peu fréquent). Si le paramètre est vide (`"path": [ { "prefix": "" } ],`), la cible correspond à tous les objets enregistrés n'importe où dans le conteneur actuel. Vous pouvez inclure jusqu'à 10 chemins de `prefix` dans une même règle.
  - `wildcard` - Vous définissez le chemin d'accès à `wildcard` si vous souhaitez transférer des objets spécifiques dans la classe de stockage IA (accès peu fréquent) en fonction du nom et/ou du type de fichier. Vous pouvez utiliser un ou plusieurs caractères génériques, représentés par un astérisque (*). Chaque caractère générique représente n'importe quelle combinaison de zéro caractère ou plus. Par exemple, `"path": [ {"wildcard": "Football/*.ts"} ],`

s'applique à tous les fichiers du dossier `Football` qui correspondent au modèle `*.ts` (tels que nomfichier.ts, nomfichier1.ts et nomfichier123456.ts). Vous pouvez inclure jusqu'à 10 chemins de `wildcard` dans une même règle.

- `days_since_create` : toujours défini sur `"numeric": [">=" , 30]`.
- `action` : toujours défini sur `ARCHIVE`.

exemple

Par exemple, un conteneur nommé `LiveEvents` comporte quatre sous-dossiers : `Football`, `Baseball`, `Basketball` et `AwardsShow`. La stratégie de cycle de vie des objets attribuée au dossier `LiveEvents` peut se présenter comme suit :

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

```
            },
            "action": "EXPIRE"
        },
        {
            "definition": {
                "path": [ { "wildcard": "Football/*.ts" }  ],
                "days_since_create": [
                    {"numeric": [">" , 20]}
                ]
            },
            "action": "EXPIRE"
        },
        {
            "definition": {
                "path": [
                    {"wildcard": "Football/index*.m3u8"}
                ],
                "seconds_since_create": [
                    {"numeric": [">" , 15]}
                ]
            },
            "action": "EXPIRE"
        },
        {
                "definition": {
                    "path": [
                        {"prefix": "Program/"}
                    ],
                    "days_since_create": [
                        {"numeric": [">=" , 30]}
                    ]
                },
                "action": "ARCHIVE"
            }
    ]
}
```

La stratégie précédente spécifie les éléments suivants :

- La première règle demande à AWS MediaStore Elemental de supprimer les objets stockés dans `LiveEvents/Football` le dossier et dans le dossier lorsqu'ils `LiveEvents/Baseball` datent de plus de 28 jours.

- La deuxième règle demande au service de supprimer les objets qui sont stockés dans le dossier `LiveEvents/AwardsShow` depuis plus de 15 jours.
- La troisième règle demande au service de supprimer les objets qui sont stockés n'importe où dans le conteneur `LiveEvents` depuis plus de 40 jours. Cette règle s'applique aux objets stockés directement dans le conteneur `LiveEvents`, ainsi qu'aux objets stockés dans les quatre sous-dossiers du conteneur.
- La quatrième règle indique au service de supprimer les objets du dossier `Football` qui correspondent au modèle `*.ts` lorsqu'ils sont plus anciens que 20 jours.
- La cinquième règle demande au service de supprimer les objets du `Football` dossier qui correspondent au modèle `index*.m3u8` après qu'ils datent de plus de 15 secondes. MediaStore supprime ces fichiers 16 secondes après leur placement dans le conteneur.
- La sixième règle indique au service de déplacer les objets du dossier `Program` vers la classe de stockage IA lorsqu'ils ont atteint 30 jours.

Pour plus d'exemples de stratégie de cycle de vie des objets, reportez-vous à la section Exemples de stratégie de cycle de vie des objets.

Ajout d'une stratégie de cycle de vie des objets à un conteneur

Une stratégie de cycle de vie des objets vous permet de spécifier pendant combien de temps stocker vos objets dans un conteneur. Vous définissez une date d'expiration, et après cette date, AWS Elemental MediaStore supprime les objets. Il faut jusqu'à 20 minutes pour que le service applique la nouvelle stratégie au conteneur.

Pour plus d'informations sur l'élaboration d'une stratégie de cycle de vie, consultez Composants d'une stratégie de cycle de vie des objets.

> **Note**
>
> Pour les règles de suppression d'objet (les objets expirent en quelques jours), il peut y avoir un léger décalage entre l'expiration d'un objet et la suppression de l'objet. Toutefois, les modifications relatives à la facturation se produisent dès que l'objet expire. Par exemple, si une règle de cycle de vie spécifie 10 `days_since_create`, le compte n'est pas facturé pour l'objet une fois que celui-ci a 10 jours, même s'il n'est pas encore supprimé.

Pour ajouter une stratégie de cycle de vie des objets (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur pour lequel vous souhaitez créer une stratégie de cycle de vie d'objet.

   La page des détails du conteneur s'affiche.
3. Dans la section Object lifecycle policy (Stratégie de cycle de vie d'objet), choisissez Create object lifecycle policy (Créer une stratégie de cycle de vie d'objet).
4. Insérez la stratégie au format JSON, puis choisissez Save (Enregistrer).

Pour ajouter une stratégie de cycle de vie des objets (AWS CLI)

1. Créez un fichier qui définit la stratégie de cycle de vie des objets :

```
{
    "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. Dans le AWS CLI, utilisez la `put-lifecycle-policy` commande :

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

Cette commande ne renvoie aucune valeur. Le service attache la stratégie spécifiée au conteneur.

Affichage d'une stratégie de cycle de vie des objets

Une stratégie de cycle de vie des objets spécifie la durée pendant laquelle des objets doivent être stockés dans un conteneur.

Pour afficher une stratégie de cycle de vie des objets (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur pour lequel vous souhaitez afficher la stratégie de cycle de vie des objets.

   La page des détails du conteneur s'affiche, avec la stratégie de cycle de vie des objets dans la section Object lifecycle policy (Stratégie de cycle de vies des objets).

Pour afficher une stratégie de cycle de vie des objets (AWS CLI)

- Dans le AWS CLI, utilisez la `get-lifecycle-policy` commande :

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

L'exemple suivant illustre la valeur de retour :

```
{
    "LifecyclePolicy": "{
        "rules": [
            {
                "definition": {
                    "path": [
                        {"prefix": "Football/"},
```

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

Modification d'une stratégie de cycle de vie des objets

Vous ne pouvez pas modifier une stratégie de cycle de vie des objets existante. Par contre, vous pouvez modifier une stratégie existante en chargeant une stratégie de remplacement. Il faut jusqu'à 20 minutes pour que le service applique la stratégie mise à jour au conteneur.

Pour modifier une stratégie de cycle de vie des objets (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur pour lequel vous souhaitez modifier la stratégie de cycle de vie des objets.

   La page des détails du conteneur s'affiche.
3. Dans la section Object lifecycle policy (Stratégie de cycle de vie des objets), choisissez Edit object lifecycle policy (Modifier une stratégie de cycle de vie des objets).
4. Modifiez la stratégie, puis choisissez Save (Enregistrer).

Pour modifier une stratégie de cycle de vie des objets (AWS CLI)

1. Créez un fichier qui définit la stratégie de cycle de vie des objets mise à jour :

```
{
    "rules": [
         {
            "definition": {
                "path": [
                    {"prefix": "Football/"},
```

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

2. Dans le AWS CLI, utilisez la `put-lifecycle-policy` commande :

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

Cette commande ne renvoie aucune valeur. Le service associe la stratégie spécifiée au conteneur, en remplaçant la stratégie précédente.

Suppression d'une stratégie de cycle de vie des objets

Lorsque vous supprimez une stratégie de cycle de vie d'objet, il faut jusqu'à 20 minutes pour que le service applique la modification au conteneur.

Pour supprimer une stratégie de cycle de vie des objets (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur pour lequel vous souhaitez supprimer la stratégie de cycle de vie des objets.

   La page des détails du conteneur s'affiche.
3. Dans la section Object lifecycle policy (Stratégie de cycle de vie des objets), choisissez Delete lifecycle policy (Supprimer une stratégie de cycle de vie des objets).
4. Choisissez Continue (Continuer) pour confirmer, puis Save (Enregistrer).

Pour supprimer une stratégie de cycle de vie des objets (AWS CLI)

- Dans le AWS CLI, utilisez la `delete-lifecycle-policy` commande :

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

Cette commande ne renvoie aucune valeur.

Exemples de stratégie de cycle de vie des objets

Les exemples suivants présentent des stratégies de cycle de vie des objets.

Rubriques

- Exemple de stratégie de cycle de vie des objets : expiration au bout de quelques secondes
- Exemple de stratégie de cycle de vie des objets : expiration au bout de quelques jours
- Exemple de stratégie de cycle de vie des objets : transfert vers une classe de stockage IA (accès peu fréquent)
- Exemple de stratégie de cycle de vie des objets : règles multiples
- Exemple de stratégie de cycle de vie des objets : conteneur vide

Exemple de stratégie de cycle de vie des objets : expiration au bout de quelques secondes

La politique suivante indique que les objets MediaStore qui répondent à tous les critères suivants sont supprimés :

- L'objet a été ajouté au conteneur après l'entrée en vigueur de la stratégie.
- L'objet est stocké dans le dossier `Football`.
- L'objet a une extension de fichier `m3u8`.
- L'objet est dans le conteneur depuis plus de 20 secondes.

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

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

Exemple de stratégie de cycle de vie des objets : expiration au bout de quelques jours

La politique suivante indique que les objets MediaStore qui répondent à tous les critères suivants sont supprimés :

- L'objet est stocké dans le dossier `Program`
- L'objet a une extension de fichier `ts`
- L'objet est dans le conteneur depuis plus de 5 jours

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

Exemple de stratégie de cycle de vie des objets : transfert vers une classe de stockage IA (accès peu fréquent)

La politique suivante indique que les MediaStore objets sont déplacés vers la classe de stockage à accès peu fréquent (IA) lorsqu'ils ont 30 jours. Les objets stockés dans la classe de stockage IA

(accès peu fréquent) ont des taux de stockage et d'extraction différents de ceux stockés dans la classe de stockage standard.

Le champ `days_since_create` doit être défini sur `"numeric": [">=" ,30]`.

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

Exemple de stratégie de cycle de vie des objets : règles multiples

La politique suivante indique qu'il MediaStore effectue les opérations suivantes :

- Déplacement des objets stockés dans le dossier `AwardsShow` vers la classe de stockage IA (accès peu fréquent) après 30 jours
- Suppression des objets ayant une extension de fichier `m3u8` et stockés dans le dossier `Football` après 20 secondes
- Suppression des objets stockés dans le dossier `April` après 10 jours
- Suppression des objets ayant une extension de fichier `ts` et stockés dans le dossier`Program` après 5 jours

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

Exemple de stratégie de cycle de vie des objets : conteneur vide

La politique de cycle de vie des objets suivante indique MediaStore que tous les objets du conteneur, y compris les dossiers et les sous-dossiers, sont supprimés un jour après leur ajout au conteneur. Si le conteneur contient des objets avant l'application de cette politique, il MediaStore supprime les objets un jour après l'entrée en vigueur de la politique. Il faut jusqu'à 20 minutes pour que le service applique la nouvelle stratégie au conteneur.

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

Politiques relatives aux métriques dans AWS Elemental MediaStore

Pour chaque conteneur, vous pouvez ajouter une politique de métriques afin de permettre à AWS Elemental MediaStore d'envoyer des métriques à Amazon. CloudWatch Jusqu'à 20 minutes peuvent s'écouler avant que la nouvelle stratégie prenne effet. Pour une description de chaque MediaStore métrique, voirMediaStore métriques.

Une stratégie de métriques contient les éléments suivants :

- Un paramètre permettant d'activer ou de désactiver les métriques au niveau du conteneur.
- De zéro à cinq règles qui activent les métriques au niveau de l'objet. Si la stratégie contient des règles, chaque règle doit inclure les deux éléments suivants :
  - Un groupe d'objets qui définit les objets à inclure dans le groupe. La définition peut être un chemin d'accès ou un nom de fichier, mais elle ne peut pas contenir plus de 900 caractères. Les caractères valides sont : a-z, A-Z, 0-9, _ (trait de soulignement), = (égal), : (deux-points),.

(point), - (trait d'union), ~ (tilde),/(barre oblique) et * (astérisque). Les caractères génériques (*) sont acceptés.

- Un nom de groupe d'objets qui vous permet de faire référence au groupe d'objets. Ce nom ne peut pas contenir plus de 30 caractères. Les caractères valides sont : a-z, A-Z, 0-9 et le trait d'union (-).

Si un objet correspond à plusieurs règles, CloudWatch affiche un point de données pour chaque règle correspondante. Par exemple, si un objet correspond à deux règles nommées `rule1` et `rule2` CloudWatch affiche deux points de données pour ces règles. Le premier a la dimension `ObjectGroupName=rule1`, et le second la dimension `ObjectGroupName=rule2`.

Rubriques

- Ajout d'une stratégie de métriques
- Affichage d'une stratégie de métriques
- Modification d'une stratégie de métriques
- Exemples de stratégies de métriques

Ajout d'une stratégie de métriques

Une politique de métriques contient des règles qui dictent les métriques qu'AWS MediaStore Elemental envoie à Amazon. CloudWatch Pour obtenir des exemples de stratégies de métriques, consultez Exemples de stratégies de métriques.

Pour ajouter une stratégie de métriques (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur auquel vous souhaitez ajouter une stratégie de métriques.

   La page des détails du conteneur s'affiche.
3. Dans la section Metric policy (Stratégie de métriques), choisissez Create metric policy (Créer une stratégie de métriques).
4. Insérez la stratégie au format JSON, puis choisissez Save (Enregistrer).

Affichage d'une stratégie de métriques

Vous pouvez utiliser la console ou le AWS CLI pour consulter la politique métrique d'un conteneur.

Pour afficher une stratégie de métriques (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur.

   La page des détails du conteneur s'affiche. La stratégie s'affiche dans la section Metric policy (Stratégie de métriques).

Modification d'une stratégie de métriques

Une politique de métriques contient des règles qui dictent les métriques qu'AWS MediaStore Elemental envoie à Amazon. CloudWatch Lorsque vous modifiez une stratégie de métriques existante, il peut s'écouler jusqu'à 20 minutes avant que les modifications prennent effet. Pour obtenir des exemples de stratégies de métriques, consultez Exemples de stratégies de métriques.

Pour modifier une stratégie de métriques (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur.
3. Dans la section Metric policy (Stratégie de métriques), choisissez Edit metric policy (Modifier une stratégie de métriques).
4. Effectuez les modifications appropriées, puis choisissez Save (Enregistrer).

Exemples de stratégies de métriques

Les exemples suivants illustrent les stratégies de métriques créées pour différents cas d'utilisation.

Rubriques

- Exemple de stratégie de métriques : Métriques au niveau du conteneur
- Exemple de stratégie de métriques : Métriques au niveau du chemin d'accès
- Exemple de stratégie de métriques : Métriques au niveau du conteneur et du chemin d'accès
- Exemple de stratégie de métriques : Métriques au niveau du chemin d'accès à l'aide de caractères génériques

- Exemple de stratégie de métriques : Métriques au niveau du chemin avec chevauchement de règles

Exemple de stratégie de métriques : Métriques au niveau du conteneur

Cet exemple de politique indique qu'AWS Elemental MediaStore doit envoyer des métriques CloudWatch à Amazon au niveau du conteneur. Par exemple, cela inclut la métrique `RequestCount` qui compte le nombre de demandes `Put` adressées au conteneur. Vous pouvez également définir cette valeur sur `DISABLED`.

Comme cette politique ne contient aucune règle, MediaStore elle n'envoie pas de métriques au niveau du chemin. Par exemple, vous ne pouvez pas voir combien de demandes `Put` ont été adressées à un dossier donné de ce conteneur.

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

Exemple de stratégie de métriques : Métriques au niveau du chemin d'accès

Cet exemple de politique indique qu'AWS Elemental ne MediaStore doit pas envoyer de métriques CloudWatch à Amazon au niveau du conteneur. En outre, MediaStore doit envoyer les métriques des objets dans deux dossiers spécifiques : `baseball/saturday` et `football/saturday`. Les métriques pour les demandes MediaStore sont les suivantes :

- Les demandes adressées au `baseball/saturday` dossier ont une CloudWatch dimension de`ObjectGroupName=baseballGroup`.
- Les demandes adressées au dossier `football/saturday` ont une dimension `ObjectGroupName=footballGroup`.

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

```
        "ObjectGroup": "football/saturday",
        "ObjectGroupName": "footballGroup"
      }
   ]
}
```

Exemple de stratégie de métriques : Métriques au niveau du conteneur et du chemin d'accès

Cet exemple de politique indique qu'AWS Elemental MediaStore doit envoyer des métriques CloudWatch à Amazon au niveau du conteneur. En outre, MediaStore doit envoyer des métriques pour les objets se trouvant dans deux dossiers spécifiques : `baseball/saturday` et`football/saturday`. Les métriques pour les demandes MediaStore sont les suivantes :

- Les demandes adressées au `baseball/saturday` dossier ont une CloudWatch dimension de`ObjectGroupName=baseballGroup`.
- Les demandes adressées au `football/saturday` dossier ont une CloudWatch dimension`ObjectGroupName=footballGroup`.

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

Exemple de stratégie de métriques : Métriques au niveau du chemin d'accès à l'aide de caractères génériques

Cet exemple de politique indique qu'AWS Elemental MediaStore doit envoyer des métriques CloudWatch à Amazon au niveau du conteneur. En outre, MediaStore doit également envoyer des métriques pour les objets en fonction de leur nom de fichier. Un caractère générique indique que les

objets peuvent être stockés n'importe où dans le conteneur et posséder n'importe quel nom de fichier, tant que celui-ci se termine par une extension `.m3u8`

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

Exemple de stratégie de métriques : Métriques au niveau du chemin avec chevauchement de règles

Cet exemple de politique indique qu'AWS Elemental MediaStore doit envoyer des métriques CloudWatch à Amazon au niveau du conteneur. En outre, MediaStore doit envoyer des métriques pour deux dossiers : `sports/football/saturday` et`sports/football`.

Les métriques relatives aux MediaStore demandes adressées au `sports/football/saturday` dossier ont une CloudWatch dimension de`ObjectGroupName=footballGroup1`. Étant donné que les objets stockés dans le dossier `sports/football` correspondent aux deux règles, CloudWatch affiche deux points de données pour ces objets : l'un avec une dimension `ObjectGroupName=footballGroup1` et le second avec une dimension `ObjectGroupName=footballGroup2`.

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

Dossiers dans AWS Elemental MediaStore

Les dossiers sont divisés au sein d'un conteneur. Utilisez des dossiers pour subdiviser votre conteneur de la même façon que pour créer des sous-dossiers pour diviser un dossier dans un système de fichiers. Vous pouvez créer jusqu'à 10 niveaux de dossiers (y compris le conteneur lui-même).

Les dossiers sont facultatifs, vous pouvez choisir de charger vos objets directement dans un conteneur au lieu d'un dossier. Toutefois, les dossiers sont un moyen facile d'organiser vos objets.

Pour charger un objet dans un dossier, vous spécifiez le chemin d'accès au dossier. Si le dossier existe déjà, AWS Elemental y MediaStore stocke l'objet. Si le dossier n'existe pas, le service le crée, puis stocke l'objet dans le dossier.

Supposons, par exemple, que vous ayez un conteneur nommé `movies` et que vous téléchargiez un fichier nommé `mlaw.ts` avec son chemin`premium/canada`. AWS Elemental MediaStore stocke l'objet dans le sous-dossier canada sous le dossier premium. Si aucun dossier n'existe, le service crée le dossier `premium` et le sous-dossier `canada`, puis stocke votre objet dans le sous-dossier `canada`. Si vous spécifiez uniquement le conteneur `movies` (sans chemin d'accès), le service stocke l'objet directement dans le conteneur.

AWS Elemental supprime MediaStore automatiquement un dossier lorsque vous supprimez le dernier objet de ce dossier. Le service supprime également les dossiers vides au-dessus de ce dossier. Par exemple, admettons que vous disposez d'un dossier nommé premium qui ne contient aucun fichier mais un sous-dossier nommé `canada`. Le sous-dossier `canada` contient un seul fichier nommé `mlaw.ts`. Si vous supprimez le fichier `mlaw.ts`, le service supprime les dossiers `premium` et `canada`. Cette suppression automatique s'applique uniquement aux dossiers. Le service ne supprime pas les conteneurs vides.

Rubriques

- Règles des noms de dossier
- Création d'un dossier
- Suppression d'un dossier

Règles des noms de dossier

Lorsque vous choisissez un nom pour votre dossier, n'oubliez pas les points suivants :

- Le nom ne peut contenir que les caractères suivants : lettres majuscules (A-Z), lettres minuscules (a-z), chiffres (0-9), points (.), tirets (-), tildes (~), traits de soulignement (_), signes égaux (=) et deux-points (:).
- Le nom doit comporter au moins un caractère. Les noms de dossiers vides (tels que`folder1//folder3/`) ne sont pas autorisés.
- Les noms sont sensibles à la casse. Par exemple, un dossier peut être nommé `myFolder` et un dossier peut être nommé `myfolder` dans le même conteneur ou dossier car ces noms sont uniques.
- Le nom doit être unique seulement dans leur dossier ou conteneur parent. Par exemple, vous pouvez créer un dossier nommé `myfolder` dans deux différents conteneurs : `movies/myfolder` et `sports/myfolder`.
- Le nom peut être le même que celui du conteneur parent.
- Le dossier ne peut pas être renommé après avoir été créé.

Création d'un dossier

Vous pouvez créer des dossiers lorsque vous chargez des objets. Pour charger un objet dans un dossier, vous spécifiez le chemin d'accès au dossier. Si le dossier existe déjà, AWS Elemental y MediaStore stocke l'objet. Si le dossier n'existe pas, le service le crée, puis stocke l'objet dans le dossier.

Pour de plus amples informations, veuillez consulter the section called "Chargement d'un objet".

Suppression d'un dossier

Vous pouvez supprimer des dossiers uniquement si le dossier est vide ; vous ne pouvez pas supprimer des dossiers contenant des objets.

AWS Elemental supprime MediaStore automatiquement un dossier lorsque vous supprimez le dernier objet de ce dossier. Le service supprime également les dossiers vides au-dessus de ce dossier. Par exemple, admettons que vous disposez d'un dossier nommé `premium` qui ne contient aucun fichier mais un sous-dossier nommé `canada`. Le sous-dossier `canada` contient un seul fichier

nommé `mlaw.ts`. Si vous supprimez le fichier `mlaw.ts`, le service supprime les dossiers `premium` et `canada`. Cette suppression automatique s'applique uniquement aux dossiers. Le service ne supprime pas les conteneurs vides.

Pour de plus amples informations, veuillez consulter Suppression d'un objet.

Objets dans AWS Elemental MediaStore

Les MediaStore actifs AWS Elemental sont appelés objets. Vous pouvez charger un objet dans un conteneur ou dans un dossier du conteneur.

Dans MediaStore, vous pouvez charger, télécharger et supprimer des objets :

- Charger – Ajouter un objet à un conteneur ou un dossier. Cette action diffère de la création d'objet. Vous devez créer vos objets localement avant de pouvoir les télécharger sur MediaStore.
- Télécharger — Copiez un objet depuis MediaStore un autre emplacement. Cela ne supprime pas l'objet de MediaStore.
- Supprimer – Supprimer complètement un objet d' MediaStore. Vous pouvez supprimer les objets individuellement, ou vous pouvez ajouter une stratégie de cycle de vie des objets pour supprimer automatiquement les objets au sein d'un conteneur après une durée spécifiée.

MediaStore accepte tous les types de fichiers.

Rubriques

- Chargement d'un objet
- Affichage d'une liste d'objets
- Affichage des détails d'un objet
- Téléchargement d'un objet
- Suppression d'objets

Chargement d'un objet

Vous pouvez charger des objets dans un conteneur ou dans un dossier d'un conteneur. Pour charger un objet dans un dossier, vous spécifiez le chemin d'accès au dossier. Si le dossier existe déjà, AWS Elemental y MediaStore stocke l'objet. Si le dossier n'existe pas, le service le crée, puis stocke l'objet dans le dossier. Pour plus d'informations sur les dossiers, consultez Dossiers dans AWS Elemental MediaStore.

Vous pouvez utiliser la MediaStore console ou le AWS CLI pour télécharger des objets.

MediaStore prend en charge le transfert fragmenté d'objets, ce qui réduit la latence en rendant un objet disponible au téléchargement alors qu'il est encore en cours de téléchargement. Pour utiliser

cette fonctionnalité, définissez la disponibilité de chargement de l'objet sur `streaming`. Vous pouvez définir la valeur de cet en-tête lorsque vous chargez l'objet à l'aide de l'API. Si vous ne spécifiez pas cet en-tête dans votre demande, MediaStore attribuez la valeur par défaut de `standard` pour la disponibilité du téléchargement de l'objet.

Les tailles d'objet ne peuvent pas dépasser 25 Mo pour une disponibilité de chargement standard ni 10 Mo pour une disponibilité de chargement en streaming.

> **Note**
>
> Les noms de fichiers d'objets ne peuvent contenir que des lettres, des chiffres, des points (.), des traits de soulignement (_), des tildes (~), des traits d'union (-), des signes égaux (=) et des deux-points (:).

Pour charger un objet (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur. Le volet des détails du conteneur s'affiche.
3. Choisissez Upload object (Charger un objet).
4. Pour Target path (Chemin d'accès cible), saisissez un chemin pour les dossiers. Par exemple, `premium/canada`. Si l'un des dossiers du chemin d'accès que vous spécifiez n'existe pas encore, le service le crée automatiquement.
5. Dans la section Object (Objet), choisissez Browse (Parcourir).
6. Naviguez jusqu'au dossier approprié et choisissez un objet à charger.
7. Choisissez Open (Ouvrir), puis Upload (Charger).

> **Note**
>
> Si un fichier du même nom existe déjà dans le dossier sélectionné, le service remplace le fichier d'origine par le fichier chargé.

Pour charger un objet (AWS CLI)

- Dans le AWS CLI, utilisez la `put-object` commande. Vous pouvez également inclure l'un des paramètres suivants : `content-type`, `cache-control` (pour autoriser l'appelant à contrôler le comportement de cache de l'objet), et `path` (pour placer l'objet dans un dossier à l'intérieur du conteneur).

  > **Note**
  >
  > Une fois l'objet chargé, vous ne pouvez pas modifier le `content-type`, `cache-control` ou `path`.

```
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
```

L'exemple suivant illustre la valeur de retour :

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

Affichage d'une liste d'objets

Vous pouvez utiliser la MediaStore console AWS Elemental pour afficher les éléments (objets et dossiers) stockés au niveau supérieur d'un conteneur ou d'un dossier. Les éléments stockés dans un sous-dossier du conteneur ou du dossier actuel ne seront pas affichés. Vous pouvez utiliser le AWS CLI pour afficher la liste des objets et des dossiers d'un conteneur, quel que soit le nombre de dossiers ou de sous-dossiers qu'il contient.

Pour afficher une liste des objets dans un conteneur spécifique (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.

2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur pour lequel vous souhaitez afficher le dossier.
3. Choisissez le nom du dossier dans la liste.

   Une page des détails s'affiche, montrant tous les dossiers et objets stockés dans le dossier.

Pour afficher une liste des objets dans un dossier spécifique (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur pour lequel vous souhaitez afficher le dossier.

   Une page des détails s'affiche, montrant tous les dossiers et objets stockés dans le conteneur.

Pour afficher une liste des objets et dossiers dans un conteneur spécifique (AWS CLI)

- Dans le AWS CLI, utilisez la `list-items` commande :

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

L'exemple suivant illustre la valeur de retour :

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

```
}
```

> **Note**
>
> Les objets qui sont soumis à une règle `seconds_since_create` ne sont pas inclus dans une réponse `list-items`.

Pour afficher une liste des objets et dossiers dans un dossier spécifique (AWS CLI)

- Dans le AWS CLI, utilisez la `list-items` commande, avec le nom de dossier spécifié à la fin de la requête :

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

L'exemple suivant illustre la valeur de retour :

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

> **Note**
>
> Les objets qui sont soumis à une règle `seconds_since_create` ne sont pas inclus dans une réponse `list-items`.

Affichage des détails d'un objet

Une fois que vous avez chargé un objet, AWS Elemental MediaStore stocke des informations telles que la date de modification, la longueur du contenu, ETag (balise d'entité) et le type de contenu. Pour de plus amples informations sur l'utilisation des métadonnées d'un objet, veuillez consulter Interaction de MediaStoreavec les caches HTTP.

Pour afficher les détails d'un objet (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur pour lequel vous souhaitez afficher l'objet.
3. Si l'objet que vous souhaitez afficher se trouve dans un dossier, choisissez les noms de dossier jusqu'à ce que vous voyiez l'objet.
4. Choisissez le nom de l'objet.

   Une page des détails s'affiche, montrant les informations sur l'objet.

Pour afficher les détails d'un objet (AWS CLI)

- Dans le AWS CLI, utilisez la `describe-object` commande :

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

L'exemple suivant illustre la valeur de retour :

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

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

Téléchargement d'un objet

Vous pouvez utiliser la console pour télécharger un objet. Vous pouvez utiliser le AWS CLI pour télécharger un objet ou uniquement une partie d'un objet.

Pour télécharger un objet (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur pour lequel vous souhaitez télécharger l'objet.
3. Si l'objet que vous souhaitez télécharger se trouve dans un dossier, choisissez les noms de dossier jusqu'à ce que vous voyiez l'objet.
4. Choisissez le nom de l'objet.
5. Sur la page des détails Object (Objet), choisissez Download (Télécharger).

Pour télécharger un objet (AWS CLI)

- Dans le AWS CLI, utilisez la `get-object` commande :

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

L'exemple suivant illustre la valeur de retour :

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

Pour télécharger une partie d'un objet (AWS CLI)

- Dans le AWS CLI, utilisez la `get-object` commande et spécifiez une plage.

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

L'exemple suivant illustre la valeur de retour :

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

Suppression d'objets

AWS Elemental MediaStore propose différentes options pour supprimer des objets des conteneurs :

- Supprimer les objets individuellement. Aucun frais ne s'applique.
- Vider un conteneur pour supprimer tous les objets d'un conteneur simultanément. Étant donné que ce processus utilise des appels d'API, des frais d'API standard s'appliquent.
- Ajouter une stratégie de cycle de vie des objets pour supprimer des objets lorsqu'ils atteignent un certain âge. Aucun frais ne s'applique.

Suppression d'un objet

Vous pouvez supprimer les objets individuellement à l'aide de la console ou de l' AWS CLI. Vous pouvez également ajouter une stratégie de cycle de vie des objets pour supprimer automatiquement les objets d'un conteneur qui atteignent un certain âge, ou vider un conteneur pour supprimer tous les objets de ce conteneur.

> **Note**
>
> Lorsque vous supprimez le seul objet d'un dossier, AWS Elemental supprime MediaStore automatiquement le dossier ainsi que tous les dossiers vides situés au-dessus de ce dossier. Par exemple, admettons que vous disposez d'un dossier nommé `premium` qui ne contient aucun fichier mais un sous-dossier nommé `canada`. Le sous-dossier `canada` contient un seul fichier nommé `mlaw.ts`. Si vous supprimez le fichier `mlaw.ts`, le service supprime les dossiers `premium` et `canada`.

Pour supprimer un objet (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur comportant l'objet à supprimer.
3. Si l'objet que vous souhaitez supprimer se trouve dans un dossier, choisissez les noms de dossier jusqu'à ce que vous voyiez l'objet.
4. Choisissez l'option à gauche du nom de l'objet.
5. Sélectionnez Delete (Supprimer).

Pour supprimer un objet (AWS CLI)

- Dans le AWS CLI, utilisez la `delete-object` commande.

  Exemple :

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

  Cette commande ne renvoie aucune valeur.

Vidage d'un conteneur

Vous pouvez vider un conteneur pour supprimer tous les objets qui y sont stockés. Vous pouvez également ajouter une stratégie de cycle de vie des objets pour supprimer automatiquement

les objets d'un conteneur qui atteignent un certain âge, ou vous pouvez supprimer les objets individuellement.

Pour vider un conteneur (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.
2. Sur la page Containers (Conteneurs), choisissez l'option correspondant au conteneur que vous souhaitez vider.
3. Choisissez Empty container (Vider le conteneur). Un message de confirmation s'affiche.
4. Confirmez que vous souhaitez vider le conteneur en saisissant le nom du conteneur dans le champ de texte, puis choisissez Vide.

Sécurité dans AWS Elemental MediaStore

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS Cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de AWS conformité Programmes de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à AWS Elemental MediaStore, consultez la section AWS Services concernés par programme de conformitéAWS .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de son utilisation MediaStore. Les rubriques suivantes expliquent comment procéder à la configuration MediaStore pour atteindre vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos MediaStore ressources.

Rubriques

- Protection des données dans AWS Elemental MediaStore
- Identity and Access Management pour AWS Elemental MediaStore
- Connexion et surveillance AWS Elemental MediaStore
- Validation de conformité pour AWS Elemental MediaStore
- Résilience dans AWS Elemental MediaStore
- Sécurité de l'infrastructure dans AWS Elemental MediaStore
- Prévention du cas de figure de l'adjoint désorienté entre services

Protection des données dans AWS Elemental MediaStore

Le modèle de responsabilité AWS partagée s'applique à la protection des données dans AWS Elemental MediaStore. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée AWS et RGPD (Règlement général sur la protection des données) sur le AWS Blog de sécurité.

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou Gestion des identités et des accès AWS (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- SSL/TLS À utiliser pour communiquer avec AWS les ressources. Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section Utilisation des CloudTrail sentiers dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-3 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS disponibles, consultez Norme FIPS (Federal Information Processing Standard) 140-3.

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Nom. Cela inclut lorsque vous travaillez avec MediaStore ou d'autres Services

AWS utilisateurs de la console, de l'API ou AWS SDKs. AWS CLI Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Chiffrement des données

MediaStore chiffre les conteneurs et les objets au repos à l'aide de l'algorithme standard AES-256. Nous vous recommandons d'utiliser MediaStore pour sécuriser vos données de la manière suivante :

- Créez une politique de conteneur pour contrôler les droits d'accès à tous les dossiers et objets de ce conteneur. Pour de plus amples informations, veuillez consulter the section called "Stratégies de conteneur".
- Créez une politique de partage des ressources entre origines (CORS) pour autoriser l'accès entre origines de manière sélective à vos ressources. MediaStore Avec le CORS, vous pouvez autoriser des applications web clientes chargées dans un domaine particulier à interagir avec les ressources d'un autre domaine. Pour de plus amples informations, veuillez consulter the section called "Stratégies CORS".

Identity and Access Management pour AWS Elemental MediaStore

Gestion des identités et des accès AWS (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser MediaStore les ressources. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- Public ciblé
- Authentification par des identités
- Gestion de l'accès à l'aide de politiques
- Comment AWS Elemental MediaStore fonctionne avec IAM
- Exemples de politiques basées sur l'identité pour AWS Elemental MediaStore
- Résolution des problèmes liés à l' MediaStore identité et à l'accès à AWS Elemental

Public ciblé

La façon dont vous utilisez Gestion des identités et des accès AWS (IAM) varie en fonction de votre rôle :

- Utilisateur du service : demandez des autorisations à votre administrateur si vous ne pouvez pas accéder aux fonctionnalités (voir Résolution des problèmes liés à l' MediaStore identité et à l'accès à AWS Elemental)
- Administrateur du service : déterminez l'accès des utilisateurs et soumettez les demandes d'autorisation (voir Comment AWS Elemental MediaStore fonctionne avec IAM)
- Administrateur IAM : rédigez des politiques pour gérer l'accès (voir Exemples de politiques basées sur l'identité pour AWS Elemental MediaStore)

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en tant qu'identité fédérée à l'aide d'informations d'identification provenant d'une source d'identité telle que AWS IAM Identity Center (IAM Identity Center), d'une authentification unique ou d'informations d'identification. Google/Facebook Pour plus d'informations sur la connexion, consultez Connexion à votre Compte AWS dans le Guide de l'utilisateur Connexion à AWS .

Pour l'accès par programmation, AWS fournit un SDK et une CLI pour signer les demandes de manière cryptographique. Pour plus d'informations, consultez Signature AWS Version 4 pour les demandes d'API dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une seule identité de connexion appelée utilisateur Compte AWS root qui dispose d'un accès complet à toutes Services AWS les ressources. Il est vivement déconseillé d'utiliser l'utilisateur racine pour vos tâches quotidiennes. Pour les tâches qui requièrent des informations d'identification de l'utilisateur racine, consultez Tâches qui requièrent les informations d'identification de l'utilisateur racine dans le Guide de l'utilisateur IAM.

Identité fédérée

Il est recommandé d'obliger les utilisateurs humains à utiliser la fédération avec un fournisseur d'identité pour accéder à Services AWS l'aide d'informations d'identification temporaires.

Une identité fédérée est un utilisateur provenant de votre annuaire d'entreprise, de votre fournisseur d'identité Web ou Directory Service qui y accède à Services AWS l'aide d'informations d'identification provenant d'une source d'identité. Les identités fédérées assument des rôles qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Pour plus d'informations, consultez Qu'est-ce que IAM Identity Center ? dans le Guide de l'utilisateur AWS IAM Identity Center .

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité qui dispose d'autorisations spécifiques pour une seule personne ou application. Nous vous recommandons d'utiliser ces informations d'identification temporaires au lieu des utilisateurs IAM avec des informations d'identification à long terme. Pour plus d'informations, voir Exiger des utilisateurs humains qu'ils utilisent la fédération avec un fournisseur d'identité pour accéder à AWS l'aide d'informations d'identification temporaires dans le guide de l'utilisateur IAM.

Les groupes IAM spécifient une collection d'utilisateurs IAM et permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Pour plus d'informations, consultez la section Cas d'utilisation pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Rôles IAM

Un rôle IAM est une identité dotée d'autorisations spécifiques qui fournit des informations d'identification temporaires. Vous pouvez assumer un rôle en passant d'un rôle d'utilisateur à un rôle IAM (console) ou en appelant une opération d' AWS API AWS CLI ou d'API. Pour plus d'informations, consultez Méthodes pour endosser un rôle dans le Guide de l'utilisateur IAM.

Les rôles IAM sont utiles pour l'accès des utilisateurs fédérés, les autorisations temporaires des utilisateurs IAM, les accès entre comptes, les accès entre services et pour les applications exécutées sur Amazon. EC2 Pour plus d'informations, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Gestion de l'accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique définit les autorisations lorsqu'elles sont associées à une identité ou à une ressource. AWS évalue ces politiques lorsqu'un directeur fait une demande. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations les documents de politique JSON, consultez Vue d'ensemble des politiques JSON dans le Guide de l'utilisateur IAM.

À l'aide de politiques, les administrateurs précisent qui a accès à quoi en définissant quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Un administrateur IAM crée des politiques IAM et les ajoute aux rôles, que les utilisateurs peuvent ensuite assumer. Les politiques IAM définissent les autorisations quelle que soit la méthode que vous utilisez pour exécuter l'opération.

Politiques basées sur l'identité

Les stratégies basées sur l'identité sont des documents de stratégie d'autorisations JSON que vous attachez à une identité (utilisateur, groupe ou rôle). Ces politiques contrôlent les actions que peuvent exécuter ces identités, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Définition d'autorisations IAM personnalisées avec des politiques gérées par le client dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être des politiques intégrées (intégrées directement dans une seule identité) ou des politiques gérées (politiques autonomes associées à plusieurs identités). Pour découvrir comment choisir entre des politiques gérées et en ligne, consultez Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Les exemples incluent les politiques de confiance de rôle IAM et les politiques de compartiment Amazon S3. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Vous devez spécifier un principal dans une politique basée sur les ressources.

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

Autres types de politique

AWS prend en charge des types de politiques supplémentaires qui peuvent définir les autorisations maximales accordées par les types de politiques les plus courants :

- Limites d'autorisations : une limite des autorisations définit le nombre maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM. Pour plus d'informations, consultez Limites d'autorisations pour des entités IAM dans le Guide de l'utilisateur IAM.
- Politiques de contrôle des services (SCPs) — Spécifiez les autorisations maximales pour une organisation ou une unité organisationnelle dans AWS Organizations. Pour plus d'informations, veuillez consulter Politiques de contrôle de service du Guide de l'utilisateur AWS Organizations .
- Politiques de contrôle des ressources (RCPs) : définissez le maximum d'autorisations disponibles pour les ressources de vos comptes. Pour plus d'informations, voir Politiques de contrôle des ressources (RCPs) dans le guide de AWS Organizations l'utilisateur.
- Politiques de session : politiques avancées que vous passez en tant que paramètre lorsque vous créez par programmation une session temporaire pour un rôle ou un utilisateur fédéré. Pour plus d'informations, consultez Politiques de session dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de l'utilisateur IAM.

Comment AWS Elemental MediaStore fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à MediaStore, découvrez les fonctionnalités IAM disponibles. MediaStore

Fonctionnalités IAM que vous pouvez utiliser avec AWS Elemental MediaStore

Fonctionnalité IAM	MediaStore soutien
Politiques basées sur l'identité	Oui
Politiques basées sur les ressources	Oui
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition de politique (spécifiques au service)	Oui
ACLs	Non
ABAC (identifications dans les politiques)	Partielle
Informations d'identification temporaires	Oui
Autorisations de principal	Oui
Rôles de service	Oui
Rôles liés à un service	Non

Pour obtenir une vue d'ensemble de la façon dont MediaStore les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez la section AWS Services compatibles avec IAM dans le Guide de l'utilisateur IAM.

Politiques basées sur l'identité pour MediaStore

Prend en charge les politiques basées sur l'identité : oui

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Définition d'autorisations IAM personnalisées avec des politiques gérées par le client dans le Guide de l'utilisateur IAM.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

Exemples de politiques basées sur l'identité pour MediaStore

Pour consulter des exemples de politiques MediaStore basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour AWS Elemental MediaStore

Politiques basées sur les ressources au sein de MediaStore

Prend en charge les politiques basées sur les ressources : oui

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Par exemple, les politiques de confiance de rôle IAM et les politiques de compartiment Amazon S3 sont des politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. Pour plus d'informations, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

> **Note**
>
> MediaStore prend également en charge les politiques de conteneur qui définissent les entités principales (comptes, utilisateurs, rôles et utilisateurs fédérés) qui peuvent effectuer des actions sur le conteneur. Pour de plus amples informations, veuillez consulter Stratégies de conteneur.

Actions politiques pour MediaStore

Prend en charge les actions de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des MediaStore actions, consultez la section Actions définies par AWS Elemental MediaStore dans le Service Authorization Reference.

Les actions de politique en MediaStore cours utilisent le préfixe suivant avant l'action :

```
mediastore
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

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

Pour consulter des exemples de politiques MediaStore basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour AWS Elemental MediaStore

Ressources politiques pour MediaStore

Prend en charge les ressources de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Pour les

actions qui ne sont pas compatibles avec les autorisations de niveau ressource, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

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

Pour consulter la liste des types de MediaStore ressources et leurs caractéristiques ARNs, consultez la section Ressources définies par AWS Elemental MediaStore dans le Service Authorization Reference. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par AWS Elemental MediaStore.

La ressource MediaStore conteneur possède l'ARN suivant :

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

Pour plus d'informations sur le format de ARNs, consultez Amazon Resource Names (ARNs) et AWS Service Namespaces.

Par exemple, pour spécifier le conteneur `AwardsShow` dans votre instruction, utilisez l'ARN suivant :

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

Clés de conditions de politique pour MediaStore

Prend en charge les clés de condition de politique spécifiques au service : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` indique à quel moment les instructions sont exécutées en fonction de critères définis. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

Pour consulter la liste des clés de MediaStore condition, consultez la section Clés de condition pour AWS Elemental MediaStore dans le Service Authorization Reference. Pour savoir avec quelles

actions et ressources vous pouvez utiliser une clé de condition, consultez Actions définies par AWS Elemental MediaStore.

Pour consulter des exemples de politiques MediaStore basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour AWS Elemental MediaStore

ACLs dans MediaStore

Supports ACLs : Non

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

ABAC avec MediaStore

Prend en charge ABAC (identifications dans les politiques) : partiellement

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit les autorisations en fonction des attributs nommés balise. Vous pouvez associer des balises aux entités et aux AWS ressources IAM, puis concevoir des politiques ABAC pour autoriser les opérations lorsque la balise du principal correspond à la balise de la ressource.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'élément de condition d'une politique utilisant les clés de condition `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur ABAC, consultez Définition d'autorisations avec l'autorisation ABAC dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez Utilisation du contrôle d'accès par attributs (ABAC) dans le Guide de l'utilisateur IAM.

Utilisation d'informations d'identification temporaires avec MediaStore

Prend en charge les informations d'identification temporaires : oui

Les informations d'identification temporaires fournissent un accès à court terme aux AWS ressources et sont automatiquement créées lorsque vous utilisez la fédération ou que vous changez de rôle. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez la section Informations d'identification de sécurité temporaires dans IAM et Services AWS compatibles avec IAM dans le Guide de l'utilisateur IAM.

Autorisations principales interservices pour MediaStore

Prend en charge les sessions d'accès direct (FAS) : oui

Les sessions d'accès direct (FAS) utilisent les autorisations du principal appelant et Service AWS, combinées Service AWS à la demande d'envoi de demandes aux services en aval. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez la section Sessions de transmission d'accès.

Rôles de service pour MediaStore

Prend en charge les rôles de service : oui

Un rôle de service est un rôle IAM qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer un rôle de service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.

> **Warning**
>
> La modification des autorisations associées à un rôle de service peut perturber MediaStore les fonctionnalités. Modifiez les rôles de service uniquement lorsque MediaStore vous recevez des instructions à cet effet.

Rôles liés à un service pour MediaStore

Prend en charge les rôles liés à un service : non

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour plus d'informations sur la création ou la gestion des rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM. Recherchez un service dans le tableau qui inclut un `Yes` dans la colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

Exemples de politiques basées sur l'identité pour AWS Elemental MediaStore

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou modifier les ressources MediaStore. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez Création de politiques IAM (console) dans le Guide de l'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par MediaStore, y compris le format de ARNs pour chacun des types de ressources, consultez la section Actions, ressources et clés de condition pour AWS Elemental MediaStore dans le Service Authorization Reference.

Rubriques

- Bonnes pratiques en matière de politiques
- Utilisation de la console MediaStore
- Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer MediaStore des ressources dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez politiques gérées par AWS ou politiques gérées par AWS pour les activités professionnelles dans le Guide de l'utilisateur IAM.

- Accordez les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation d'IAM pour appliquer des autorisations, consultez politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.
- Utilisez des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.
- Utilisez l'Analyseur d'accès IAM pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : l'Analyseur d'accès IAM valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez Validation de politiques avec IAM Access Analyzer dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger la MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez Sécurisation de l'accès aux API avec MFA dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

Utilisation de la console MediaStore

Pour accéder à la MediaStore console AWS Elemental, vous devez disposer d'un minimum d'autorisations. Ces autorisations doivent vous permettre de répertorier et d'afficher les détails MediaStore des ressources de votre Compte AWS. Si vous créez une politique basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette politique.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API qu'ils tentent d'effectuer.

Pour garantir que les utilisateurs et les rôles peuvent toujours utiliser la MediaStore console, associez également la politique MediaStore *ConsoleAccess* ou la politique *ReadOnly* AWS gérée aux entités. Pour plus d'informations, consultez Ajout d'autorisations à un utilisateur dans le Guide de l'utilisateur IAM.

Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or AWS .

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

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

Résolution des problèmes liés à l' MediaStore identité et à l'accès à AWS Elemental

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec MediaStore IAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans MediaStore
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes MediaStore ressources

Je ne suis pas autorisé à effectuer une action dans MediaStore

Si vous recevez une erreur qui indique que vous n'êtes pas autorisé à effectuer une action, vos politiques doivent être mises à jour afin de vous permettre d'effectuer l'action.

L'exemple d'erreur suivant se produit quand l'utilisateur IAM `mateojackson` tente d'utiliser la console pour afficher des informations détaillées sur une ressource *`my-example-widget`* fictive, mais ne dispose pas des autorisations `mediastore:`*`GetWidget`* fictives.

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

Dans ce cas, la politique qui s'applique à l'utilisateur `mateojackson` doit être mise à jour pour autoriser l'accès à la ressource *`my-example-widget`* à l'aide de l'action `mediastore:`*`GetWidget`*.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez une erreur selon laquelle vous n'êtes pas autorisé à exécuter `iam:PassRole` l'action, vos stratégies doivent être mises à jour afin de vous permettre de transmettre un rôle à MediaStore.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, vous devez disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour exécuter une action dans MediaStore. Toutefois, l'action nécessite que le service ait des autorisations accordées par une fonction de service. Mary n'est pas autorisée à transmettre le rôle au service.

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

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes MediaStore ressources

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour plus d'informations, consultez les éléments suivants :

- Pour savoir si ces fonctionnalités sont prises MediaStore en charge, consultezComment AWS Elemental MediaStore fonctionne avec IAM.
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez dans le Guide de l'utilisateur IAM.

- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.
- Pour en savoir plus sur la différence entre l'utilisation des rôles et des politiques basées sur les ressources pour l'accès intercompte, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Connexion et surveillance AWS Elemental MediaStore

Cette section fournit une présentation des options de consignation et surveillance dans AWS Elemental MediaStore à des fins de sécurité. Pour plus d'informations sur la consignation et la surveillance dans MediaStore consultez Surveillance et balisage dans AWS Elemental MediaStore.

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité AWS Elemental MediaStore et des performances de vos AWS solutions. Vous devez collecter des données de surveillance provenant de toutes les parties de votre AWS solution afin de pouvoir corriger plus facilement une défaillance multipoint, le cas échéant. AWS fournit plusieurs outils pour surveiller vos MediaStore ressources et répondre aux incidents potentiels.

CloudWatch Alarmes Amazon

À l'aide d' CloudWatch alarmes, vous observez une seule métrique sur une période que vous spécifiez. Si la métrique dépasse un seuil donné, une notification est envoyée à une rubrique Amazon SNS ou à une politique d'AWS Auto Scaling. CloudWatch les alarmes n'appellent pas d'actions car elles sont dans un état particulier. L'état doit avoir changé et avoir été conservé pendant un nombre de périodes spécifié. Pour de plus amples informations, veuillez consulter Surveillance avec CloudWatch.

AWS CloudTrail journaux

CloudTrail fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans AWS Elemental MediaStore. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite MediaStore, l'adresse IP à partir de laquelle la

demande a été faite, qui a fait la demande, quand elle a été faite et des détails supplémentaires. Pour de plus amples informations, veuillez consulter Journalisation des appels d'API avec CloudTrail.

AWS Trusted Advisor

Trusted Advisor s'appuie sur les meilleures pratiques apprises en servant des centaines de milliers de AWS clients. Trusted Advisor inspecte votre environnement AWS, puis émet des recommandations lorsque des opportunités se présentent pour économiser de l'argent, améliorer la disponibilité et les performances du système ou contribuer à combler les failles de sécurité. Tous les AWS clients ont accès à cinq chèques Trusted Advisor. Les clients disposant d'un plan de support Business ou Enterprise peuvent consulter tous les Trusted Advisor chèques.

Pour de plus amples informations, veuillez consulter AWS Trusted Advisor.

Validation de conformité pour AWS Elemental MediaStore

Pour savoir si un programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de AWS conformité Programmes AWS de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir Téléchargement de rapports dans AWS Artifact .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. Pour plus d'informations sur votre responsabilité en matière de conformité lors de l'utilisation Services AWS, consultez AWS la documentation de sécurité.

Résilience dans AWS Elemental MediaStore

L'infrastructure AWS mondiale est construite autour Régions AWS de zones de disponibilité. Régions AWS fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur les zones de disponibilité Régions AWS et les zones de disponibilité, consultez la section Infrastructure AWS globale.

Outre l'infrastructure AWS mondiale, MediaStore propose plusieurs fonctionnalités pour répondre à vos besoins en matière de résilience et de sauvegarde des données.

Sécurité de l'infrastructure dans AWS Elemental MediaStore

En tant que service géré, AWS Elemental MediaStore est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section Sécurité du AWS cloud. Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section Protection de l'infrastructure dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder MediaStore via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

Prévention du cas de figure de l'adjoint désorienté entre services

Le problème de député confus est un problème de sécurité dans lequel une entité qui n'est pas autorisée à effectuer une action peut contraindre une entité plus privilégiée à le faire. En AWS, l'usurpation d'identité interservices peut entraîner la confusion des adjoints. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé et ses autorisations utilisées pour agir sur les ressources d'un autre client auxquelles on ne serait pas autorisé à accéder autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Nous recommandons d'utiliser les clés de contexte de condition `aws:SourceAccount`globale `aws:SourceArn`et les clés de contexte dans les politiques de ressources afin de limiter les autorisations qu'AWS Elemental MediaStore accorde à un autre service à la ressource. Utilisez

`aws:SourceArn` si vous souhaitez qu'une seule ressource soit associée à l'accès entre services. Utilisez `aws:SourceAccount` si vous souhaitez autoriser l'association d'une ressource de ce compte à l'utilisation interservices.

Le moyen le plus efficace de se protéger contre le problème de député confus consiste à utiliser la clé de contexte de condition globale `aws:SourceArn` avec l'ARN complet de la ressource. Si vous ne connaissez pas l'ARN complet de la ressource ou si vous spécifiez plusieurs ressources, utilisez la clé de contexte de condition globale `aws:SourceArn` avec des caractères génériques (*) pour les parties inconnues de l'ARN. Par exemple, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Si la valeur `aws:SourceArn` ne contient pas l'ID du compte, tel qu'un ARN de compartiment Amazon S3, vous devez utiliser les deux clés de contexte de condition globale pour limiter les autorisations.

La valeur de `aws:SourceArn` doit être la configuration qui MediaStore publie CloudWatch les journaux pour votre région et votre compte.

L'exemple suivant montre comment vous pouvez utiliser les touches de contexte de condition `aws:SourceAccount` globale `aws:SourceArn` et globale MediaStore pour éviter le problème de confusion des adjoints.

JSON

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

```
        "StringEquals": {
          "aws:SourceAccount": "333333333333"
        }
      }
    }
  ]
}
```

Surveillance et balisage dans AWS Elemental MediaStore

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité et des performances d'AWS Elemental MediaStore et de vos autres AWS solutions. AWS fournit les outils de surveillance suivants pour surveiller MediaStore, signaler tout problème et prendre des mesures automatiques le cas échéant :

- AWS CloudTrailcapture les appels d'API et les événements associés effectués par ou pour le compte de votre AWS compte et envoie les fichiers journaux dans un compartiment Amazon S3 que vous spécifiez. Vous pouvez identifier les utilisateurs et les comptes appelés AWS, l'adresse IP source à partir de laquelle les appels ont été effectués et la date des appels. Pour plus d'informations, consultez le Guide de l'utilisateur AWS CloudTrail.
- Amazon CloudWatch surveille vos AWS ressources et les applications que vous utilisez AWS en temps réel. Vous pouvez collecter et suivre les métriques, créer des tableaux de bord personnalisés, et définir des alarmes qui vous informent ou prennent des mesures lorsqu'une métrique spécifique atteint un seuil que vous spécifiez. Par exemple, vous pouvez CloudWatch suivre l'utilisation du processeur ou d'autres indicateurs de vos EC2 instances Amazon et lancer automatiquement de nouvelles instances en cas de besoin. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.
- Amazon CloudWatch Events fournit un flux d'événements système décrivant les modifications apportées aux AWS ressources. Généralement, les AWS services envoient des notifications d' CloudWatch événements à Events en quelques secondes, mais cela peut parfois prendre une minute ou plus. CloudWatch Les événements permettent une informatique automatisée axée sur les événements, car vous pouvez rédiger des règles qui surveillent certains événements et déclenchent des actions automatisées dans d'autres AWS services lorsque ces événements se produisent. Pour plus d'informations, consultez le guide de l'utilisateur d'Amazon CloudWatch Events.
- Amazon CloudWatch Logs vous permet de surveiller, de stocker et d'accéder à vos fichiers journaux à partir d' EC2 instances Amazon et d'autres sources. CloudTrail CloudWatch Les journaux peuvent surveiller les informations contenues dans les fichiers journaux et vous avertir lorsque certains seuils sont atteints. Vous pouvez également archiver vos données de journaux dans une solution de stockage hautement durable. Pour plus d'informations, consultez le guide de l'utilisateur d'Amazon CloudWatch Logs.

Vous pouvez également attribuer des métadonnées à vos MediaStore conteneurs sous forme de balises. Chaque balise est une étiquette qui se compose d'une clé et d'une valeur que vous définissez. Les balises peuvent faciliter la gestion, la recherche et le filtrage des ressources. Vous pouvez utiliser des balises pour organiser vos AWS ressources dans la console de AWS gestion, créer des rapports d'utilisation et de facturation pour l'ensemble de vos AWS ressources et filtrer les ressources lors des activités d'automatisation de l'infrastructure.

Rubriques

- Journalisation des appels d' MediaStore API AWS Elemental avec AWS CloudTrail
- Surveillance d'AWS Elemental MediaStore avec Amazon CloudWatch
- Balisage des ressources AWS Elemental MediaStore

Journalisation des appels d' MediaStore API AWS Elemental avec AWS CloudTrail

AWS Elemental MediaStore est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans. MediaStore CloudTrail capture un sous-ensemble d'appels d'API sous MediaStore forme d'événements, y compris les appels depuis la MediaStore console et les appels de code vers l' MediaStore API. Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris les événements pour MediaStore. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite MediaStore, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite, etc.

Pour en savoir plus CloudTrail, notamment comment le configurer et l'activer, consultez le guide de AWS CloudTrail l'utilisateur.

Rubriques

- Informations AWS Elemental MediaStore dans CloudTrail
- Exemple : entrées du fichier MediaStore journal AWS Elemental

Informations AWS Elemental MediaStore dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité événementielle prise en charge se produit dans AWS Elemental MediaStore, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. Vous pouvez consulter, rechercher et télécharger les événements récents dans votre AWS compte. Pour plus d'informations, consultez la section Affichage des événements à l'aide de l'historique des CloudTrail événements.

Pour un enregistrement continu des événements de votre AWS compte, y compris des événements pour MediaStore, créez un parcours. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un journal de suivi dans la console, il s'applique à toutes les régions AWS . Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment Amazon S3 que vous spécifiez. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d'informations, consultez les rubriques suivantes :

- Présentation de la création d'un journal d'activité
- CloudTrail Services et intégrations pris en charge
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers CloudTrail journaux de plusieurs régions et réception de fichiers CloudTrail journaux de plusieurs comptes

AWS Elemental MediaStore prend en charge la journalisation des opérations suivantes sous forme d'événements dans des fichiers CloudTrail journaux :

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

- `PutContainerPolicy`
- `PutCorsPolicy`

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été faite avec l'utilisateur root ou les informations d'identification de l'utilisateur
- Si la demande a été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré
- Si la demande a été faite par un autre AWS service

Pour plus d'informations, consultez la section Élément userIdentity CloudTrail.

Exemple : entrées du fichier MediaStore journal AWS Elemental

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande individuelle à partir d'une source quelconque et comprend des informations sur l'action demandée, la date et l'heure, les paramètres de la demande, etc. Les fichiers journaux CloudTrail ne sont pas des séries ordonnées retraçant les appels d'API publics. Ils ne suivent aucun ordre précis.

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'`CreateContainer`opération :

```
{
    "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"
}
```


Surveillance d'AWS Elemental MediaStore avec Amazon CloudWatch

Vous pouvez surveiller AWS Elemental à MediaStore l'aide d'AWS Elemental CloudWatch, qui collecte les données brutes et les traite en métriques lisibles. CloudWatch conserve les statistiques pendant 15 mois afin que vous puissiez accéder aux informations historiques et avoir une meilleure idée des performances de votre application ou service Web. Vous pouvez également définir des alarmes qui surveillent certains seuils et envoient des notifications ou prennent des mesures lorsque ces seuils sont atteints. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.

AWS fournit les outils de surveillance suivants pour surveiller MediaStore, signaler tout problème et prendre des mesures automatiques le cas échéant :

- Amazon CloudWatch Logs vous permet de surveiller, de stocker et d'accéder à vos fichiers journaux à partir de AWS services tels qu'AWS Elemental MediaStore. Vous pouvez utiliser CloudWatch les journaux pour surveiller les applications et les systèmes à l'aide des données des journaux. Par exemple, CloudWatch Logs peut suivre le nombre d'erreurs qui se produisent dans les journaux de votre application et vous envoyer une notification chaque fois que le taux d'erreurs dépasse un seuil que vous spécifiez. CloudWatch Logs utilise les données de vos journaux à des fins de surveillance, aucune modification de code n'est donc requise. Par exemple, vous pouvez surveiller les journaux des applications pour détecter des termes littéraux spécifiques (tels que ValidationException « ») ou compter le nombre de `PutObject` demandes effectuées au cours d'une période donnée. Lorsque le terme que vous recherchez est trouvé, CloudWatch Logs rapporte les données à une CloudWatch métrique que vous spécifiez. Les données des journaux sont chiffrées, pendant le transit et pendant le repos.
- Amazon CloudWatch Events fournit des événements système qui décrivent les modifications apportées aux AWS ressources, telles que MediaStore les objets. Généralement, les AWS services envoient des notifications d' CloudWatch événements à Events en quelques secondes, mais cela peut parfois prendre une minute ou plus. Vous pouvez définir des règles correspondant à des événements (tels qu'une `DeleteObject` demande) et les acheminer vers une ou plusieurs fonctions ou flux cibles. CloudWatch Events prend conscience des changements opérationnels au fur et à mesure qu'ils se produisent. En outre, CloudWatch Events répond à ces changements opérationnels et prend les mesures correctives nécessaires, en envoyant des messages pour répondre à l'environnement, en activant des fonctions, en apportant des modifications et en capturant des informations d'état.

CloudWatch Journaux

La journalisation des accès fournit des enregistrements détaillés pour les demandes effectuées vers des objets d'un conteneur. Les journaux d'accès sont utiles pour de nombreuses applications, telles que les audits de sécurité et des accès. Ils peuvent également vous aider à mieux connaître votre clientèle et à comprendre votre MediaStore facture. CloudWatch Les journaux sont classés comme suit :

- Un flux de journal est une séquence d'événements du journaux qui partagent la même source.
- Un groupe de journaux est un groupe de flux de journaux qui partagent les mêmes paramètres de conservation, de surveillance et de contrôle d’accès. Lorsque vous activez la journalisation des accès sur un conteneur, MediaStore crée un groupe de journaux portant un nom tel que`/aws/mediastore/MyContainerName`. Vous pouvez définir des groupes de journaux et spécifier les

flux à placer dans chaque groupe. Le nombre de flux de journaux pouvant appartenir à un groupe de journaux est illimité.

Par défaut, les journaux sont conservés indéfiniment et n'expirent jamais. Vous pouvez ajuster la stratégie de conservation pour chaque groupe de journaux, en gardant la conservation indéfinie, ou en choisissant une période de conservation de un jour à 10 ans.

Configuration des autorisations pour Amazon CloudWatch

Utilisez Gestion des identités et des accès AWS (IAM) pour créer un rôle permettant à AWS MediaStore Elemental d'accéder à Amazon. CloudWatch Vous devez effectuer ces étapes pour que CloudWatch les journaux soient publiés pour votre compte. CloudWatch publie automatiquement les statistiques de votre compte.

Pour autoriser MediaStore l'accès à CloudWatch

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation de la console IAM, choisissez Policies (Politiques), puis Create policy (Créer une politique).
3. Choisissez l'onglet JSON et collez la stratégie suivante :

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:CreateLogGroup"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
```

```
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/mediastore/*"
        }
    ]
}
```

Cette politique permet MediaStore de créer des groupes de journaux et des flux de journaux pour tous les conteneurs de n'importe quelle région de votre AWS compte.

4. Choisissez Examiner une politique.
5. Sur la page Examiner une stratégie, pour Nom, entrez **`MediaStoreAccessLogsPolicy`**, puis choisissez Créer une stratégie.
6. Dans le volet de navigation de la console IAM, sélectionnez Roles (Rôles), puis Create role (Créer un rôle).
7. Choisissez le type de rôle Autre compte AWS.
8. Pour ID de compte, entrez votre identifiant de AWS compte.
9. Choisissez Suivant : Autorisations.
10. Dans la zone de recherche, saisissez **`MediaStoreAccessLogsPolicy`**.
11. Activez la case à cocher en regard de votre nouvelle stratégie, puis choisissez Suivant : Balises.
12. Choisissez Suivant : Vérification pour afficher votre nouvel utilisateur.
13. Pour Nom du rôle, saisissez **`MediaStoreAccessLogs`**, puis choisissez Créer un rôle.
14. Dans le message de confirmation, choisissez le nom du rôle que vous venez de créer (**`MediaStoreAccessLogs`**).
15. Sur la page Récapitulatif du rôle, choisissez l'onglet Relations d'approbation.
16. Choisissez Modifier la relation d'approbation.
17. Dans le document de stratégie, remplacez le mandataire par le service MediaStore. Elle doit ressembler à ce qui suit :

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

L'ensemble de la stratégie doit se présenter comme suit :

JSON

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

18. Choisissez Mettre à jour la politique d'approbation.

Activation de la journalisation des accès pour un conteneur

Par défaut, AWS Elemental MediaStore ne collecte pas les journaux d'accès. Lorsque vous activez la journalisation des accès sur un conteneur, vous MediaStore transmettez à Amazon les journaux d'accès pour les objets stockés dans ce conteneur CloudWatch. Les journaux d'accès fournissent des enregistrements détaillés pour les demandes effectuées vers n'importe quel objet stocké dans le conteneur. Ces informations peuvent comprendre le type de demande, les ressources spécifiées dans la demande, ainsi que l'heure et la date du traitement de la demande.

> **Important**
>
> L'activation de la journalisation des accès sur un conteneur MediaStore n'entraîne pas de frais supplémentaires. Toutefois, les fichiers journaux qui vous sont fournis par le service augmentent les coûts de stockage habituels. (Vous pouvez supprimer les fichiers journaux à tout moment.) AWS n'évalue pas de frais de transfert de données pour la remise des fichiers journaux, mais facture des frais standard de transfert de données pour l'accès aux fichiers journaux.

Pour activer la journalisation des accès (AWS CLI)

- Dans le AWS CLI, utilisez la `start-access-logging` commande :

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

  Cette commande ne renvoie aucune valeur.

Désactivation de la journalisation des accès pour un conteneur

Lorsque vous désactivez la journalisation des accès sur un conteneur, AWS Elemental MediaStore cesse d'envoyer les journaux d'accès à Amazon. CloudWatch Ces journaux d'accès ne sont pas enregistrés et ne sont pas récupérables.

Pour désactiver la journalisation des accès (AWS CLI)

- Dans le AWS CLI, utilisez la `stop-access-logging` commande :

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

  Cette commande ne renvoie aucune valeur.

Résolution des problèmes de journalisation des accès dans AWS Elemental MediaStore

Lorsque les journaux d' MediaStore accès à AWS Elemental n'apparaissent pas sur Amazon CloudWatch, reportez-vous au tableau suivant pour connaître les causes potentielles et les solutions.

> **Note**
>
> Assurez-vous d'activer AWS CloudTrail les journaux pour faciliter le processus de dépannage.

Symptôme	Le problème peut être...	Essayez ceci...
Vous ne voyez aucun CloudTrail événement, même si CloudTrail les journaux sont activés.	Le rôle IAM n'existe pas ou il a un nom, des autorisations ou une stratégie d'approbation incorrects.	Créez un rôle avec le nom, les autorisations et la stratégie d'approbation corrects. Consultez the section called “Configuration des autorisations pour CloudWatch”.
Vous avez envoyé une demande d'API `DescribeContainer` , mais la réponse montre que le paramètre `AccessLoggingEnabled` a la valeur `False`. En outre, vous ne voyez pas d'événements CloudTrail événements pour le rôle `MediaStoreAccessLogs` qui effectue une appel `DescribeLogGroup` , `CreateLogGroup` , `DescribeLogStream` ou `CreateLogStream` réussi.	Le rôle IAM n'existe pas ou il a un nom, des autorisations ou une stratégie d'approbation incorrects.	Créez un rôle avec le nom, les autorisations et la stratégie d'approbation corrects. Consultez the section called “Configuration des autorisations pour CloudWatch”.
	La journalisation des accès n'est pas activé sur le conteneur.	Activez les journaux d'accès pour le conteneur. Consultez the section called “Activation de la journalisation des accès”.

Sur la CloudTrail console, vous voyez un événement avec une erreur de refus d'accès liée au `MediaStoreAccessLogs` rôle. L' CloudTrail événement peut inclure des lignes telles que les suivantes :
`"eventSource": "logs.amazonaws.com",`
`"errorCode": "AccessDenied",`
`"errorMessage": "User: arn:aws:sts::11112`
	Le rôle IAM ne dispose pas des autorisations appropriées pour AWS MediaStore Elemental.	Mettez à jour le rôle IAM pour qu'il dispose des autorisations et de la stratégie d'approbation appropriées. Consultez the section called “Configuration des autorisations pour CloudWatch”.

Symptôme	Le problème peut être...	Essayez ceci...
`2223333:assumed-role/MediaStoreAccessLogs/MediaStoreAccessLogsSession is not authorized to perform: logs:DescribeLogGroups on resource: arn:aws:logs:us-west-2:11112222333:log-group::log-stream:",`		
Vous ne voyez pas de journaux pour la totalité d'un ou de plusieurs conteneurs.	Votre compte a peut-être dépassé le CloudWatch quota de groupes de journaux par compte et par région. Consultez les quotas pour les groupes de journaux dans le guide de l'utilisateur Amazon CloudWatch Logs.	Sur la CloudWatch console, déterminez si votre compte a atteint le CloudWatch quota de groupes de journaux. Si nécessaire, demandez une augmentation du quota.
Certains journaux s'affichent CloudWatch, mais pas tous ceux que vous vous attendiez à voir.	Votre compte a peut-être dépassé le CloudWatch quota de transactions par seconde, par compte et par région. Consultez les quotas `PutLogEvents` dans le guide de l'utilisateur d'Amazon CloudWatch Logs.	Demandez une augmentation du quota de CloudWatch transactions par seconde, par compte et par région.

Format des journaux d'accès

Les fichiers journaux d'accès se composent d'une séquence d'enregistrements de journal au format JSON, chaque enregistrement de journal représentant une demande. L'ordre des champs dans le journal peut varier. Voici un exemple de journal constitué de deux enregistrements de journal :


```
{
  "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
}
```

La liste suivante décrit les champs de l'enregistrement des journaux :

AWSAccountId

L' AWS identifiant du compte qui a été utilisé pour effectuer la demande.

BytesReceived

Nombre d'octets dans le corps de la demande reçue par le serveur MediaStore.

BytesSent

Nombre d'octets dans le corps de la réponse envoyée par le serveur MediaStore. Cette valeur est souvent la même que celle de l'en-tête `Content-Length` inclus avec les réponses de serveur.

ContainerName

Nom du conteneur qui a reçu la demande.

ErrorCode

Le code MediaStore d'erreur (tel que`InternalServerError`). Si aucune erreur ne s'est produit, le caractère `-` s'affiche. Un code d'erreur peut s'afficher, même si le code de statut est 200 (indiquant une connexion fermée ou une erreur après que le serveur a commencé à diffuser la réponse).

ExpiresAt

Date et heure d'expiration de l'objet. Cette valeur est basée sur l'âge d'expiration défini par une transient data rule politique de cycle de vie appliquée au conteneur. La valeur correspond à une date et une heure `ISO-8601` est est basée sur l'horloge système de l'hôte ayant servi la demande. Si la politique de cycle de vie ne comporte pas de règle de données transitoires applicable à l'objet, ou si aucune politique de cycle de vie n'est appliquée au conteneur, la valeur de ce champ est. `null` Ce champ s'applique uniquement aux opérations suivantes : `PutObject``GetObject`,`DescribeObject`, et`DeleteObject`.

HTTPStatus

Code numérique du statut HTTP de la réponse.

Opération

Opération qui a été exécutée, comme `PutObject` ou `ListItems`.

Chemin

Chemin d'accès au sein du conteneur dans lequel l'objet est stocké. Si l'opération n'accepte pas le paramètre de chemin, le caractère - s'affiche.

ReceivedTime

Moment où la demande a été reçue. La valeur correspond à une date et une heure `ISO-8601` est est basée sur l'horloge système de l'hôte ayant servi la demande.

Demandeur

Amazon Resource Name (ARN) de l'utilisateur du compte qui a été utilisé pour effectuer la demande. Pour les demandes non authentifiées, cette valeur est `anonymous`. Si la demande échoue avant la fin de l'authentification, ce champ peut être manquant dans le journal. Pour des demandes de ce type, l'élément `ErrorCode` peut identifier le problème d'autorisation.

RequestID

Chaîne générée par AWS Elemental MediaStore pour identifier de manière unique chaque demande.

Source

Adresse Internet apparente du demandeur ou du mandataire de service du service AWS qui effectue l'appel. Si des proxys et pare-feu intermédiaires masquent l'adresse de la machine qui fait la demande, la valeur est null.

TotalTime

Nombre de millisecondes (ms) pendant lesquelles la demande était en cours depuis la perspective du serveur. Cette valeur est mesurée du moment où votre demande est reçue par le service jusqu'à ce que le dernier octet de la réponse soit envoyé. Cette valeur est mesurée depuis la perspective du serveur, car les mesures effectuées depuis la perspective du client sont affectées par la latence du réseau.

TurnAroundTime

Le nombre de millisecondes MediaStore passées à traiter votre demande. Cette valeur est mesurée entre la réception du dernier octets de votre demande et l'envoi du premier octet de la réponse.

L'ordre des champs dans le journal peut varier.

Les changements du statut de journalisation prennent effet au fil du temps

Les changements du statut de journalisation d'un conteneur mettent du temps à impacter la livraison des fichiers journaux. Par exemple, si vous activez la journalisation pour un conteneur, des demandes faites dans l'heure suivante peuvent être consignées, tandis que d'autres non. Si vous désactivez la journalisation pour le conteneur B, certains journaux pour l'heure suivante continuent d'être fournis, tandis que d'autres non. Dans tous les cas, les nouveaux paramètres finiront par prendre effet sans action supplémentaire de votre part.

Livraison des journaux du serveur dans la mesure du possible

Les enregistrements des journaux d'accès sont distribués dans la mesure du possible. La plupart des demandes soumises à un conteneur qui sont correctement configurées pour la journalisation se traduisent par la remise d'un enregistrement de journal. La plupart des enregistrements de journal sont distribués dans les heures qui suivent leur enregistrement, mais ils peuvent être distribués plus fréquemment.

L'exhaustivité et le timing de la journalisation des accès ne sont pas garanties. L'enregistrement d'une demande particulière peut être distribuée é longtemps après le traitement de la demande, ou ne pas être distribué du tout. Le but des journaux d'accès est de vous donner une idée de la nature du trafic dans le conteneur. La perte d'enregistrement de journal est rare, mais le journal des accès n'est pas censé tenir une comptabilité complète de toutes les demandes.

Elle découle de la fonction de journalisation des accès dans la mesure du possible avec laquelle les rapports d'utilisation disponibles sur le portail AWS (rapports de facturation et de gestion des coûts sur la Console de gestion AWS) peuvent inclure une ou plusieurs demandes qui n'apparaissent pas dans le journal des accès fourni.

Considérations en matière de programmation pour le format des journaux d'accès

De temps en temps, nous pouvons étendre le format des journaux d'accès en ajoutant de nouveaux champs. Un code qui analyse les journaux d'accès doit être écrit pour gérer les champs supplémentaires qui ne sont pas compris.

CloudWatch Événements

Amazon CloudWatch Events vous permet d'automatiser vos AWS services et de répondre automatiquement aux événements du système tels que les problèmes de disponibilité des applications ou les modifications des ressources. Vous pouvez écrire des règles simples pour

indiquer quels événements vous intéressent et les actions automatisées à effectuer quand un événement correspond à une règle.

> **Important**
>
> Généralement, les AWS services envoient des notifications d' CloudWatch événements à Events en quelques secondes, mais cela peut parfois prendre une minute ou plus.

Lorsqu'un fichier est chargé dans un conteneur ou retiré d'un conteneur, deux événements sont déclenchés successivement dans le CloudWatch service :

1. the section called “Événement de modification de l'état d'un objet”
2. the section called “Événement de modification de l'état d'un conteneur”

Pour plus d'informations sur l'abonnement à ces événements, consultez Amazon CloudWatch.

Les actions pouvant être déclenchées automatiquement sont les suivantes :

- Invoquer une fonction AWS Lambda
- Invocation de la commande Amazon EC2 Run
- Relais de l'événement à Amazon Kinesis Data Streams
- Activation d'une machine à AWS Step Functions états
- Notification d'une rubrique Amazon SNS ou d'une file d'attente AWS SMS

Voici quelques exemples d'utilisation d' CloudWatch Events avec AWS Elemental MediaStore :

- Activation d'une fonction Lambda chaque fois qu'un conteneur est créé
- Notifier un sujet Amazon SNS lorsqu'un objet est supprimé

Pour plus d'informations, consultez le guide de l'utilisateur d'Amazon CloudWatch Events.

Rubriques

- Événement de changement d'état d' MediaStore un objet AWS Elemental
- Événement de changement d'état du MediaStore conteneur AWS Elemental

Événement de changement d'état d' MediaStore un objet AWS Elemental

Cet événement est publié lors de la modification de l'état d'un objet (lorsqu'un objet a été chargé ou supprimé).

> **Note**
>
> Les objets qui expirent en raison d'une règle de données transitoire n'émettent pas d' CloudWatchévénement lorsqu'ils expirent.

Pour plus d'informations sur l'inscription à cet événement, consultez Amazon CloudWatch.

Objet mis à jour

```
{
  "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"
  }
}
```

Objet supprimé

```
{
  "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"
  }
}
```

Événement de changement d'état du MediaStore conteneur AWS Elemental

Cet événement est publié lors de la modification de l'état d'un conteneur (lorsqu'un conteneur a été ajouté ou supprimé). Pour plus d'informations sur l'inscription à cet événement, consultez Amazon CloudWatch.

Conteneur créé

```
{
  "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"
  }
}
```

Conteneur supprimé

```
{
  "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"
  }
}
```

Surveillance d'AWS Elemental à l'aide des métriques MediaStore Amazon CloudWatch

Vous pouvez surveiller AWS Elemental à MediaStore l'aide d'AWS Elemental CloudWatch, qui collecte les données brutes et les traite en métriques lisibles. CloudWatchles statistiques sont conservées pendant 15 mois afin que vous puissiez accéder aux informations historiques et avoir une meilleure idée des performances de votre application ou service Web. Vous pouvez également définir des alarmes qui surveillent certains seuils et envoient des notifications ou prennent des mesures lorsque ces seuils sont atteints. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.

Pour AWS Elemental MediaStore, vous pouvez regarder `BytesDownloaded` et vous envoyer un e-mail lorsque cette métrique atteint un certain seuil.

Pour afficher les métriques à l'aide de la CloudWatch console

Les métriques sont d'abord regroupées par espace de noms de service, puis par les différentes combinaisons de dimension au sein de chaque espace de noms.

1. Connectez-vous à la CloudWatch console Console de gestion AWS et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudwatch/.

2. Dans le panneau de navigation, sélectionnez Métriques.
3. Sous Toutes les métriques, choisissez l'espace de MediaStore noms AWS/.
4. Choisissez la dimension de métrique pour afficher les métriques. Par exemple, choisissez `Request metrics by container` pour afficher les métriques pour les différents types de demandes qui ont été envoyées au conteneur.

Pour consulter les statistiques à l'aide du AWS CLI

- À partir d'une invite de commande, utilisez la commande suivante :

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

Métriques AWS Elemental MediaStore

Le tableau suivant répertorie les métriques auxquelles AWS Elemental MediaStore envoie des données. CloudWatch

> **Note**
>
> Pour consulter les métriques, vous devez ajouter une politique de métriques au conteneur afin de MediaStore permettre l'envoi de métriques à Amazon CloudWatch.

Métrique	Description
`RequestCount`	Le nombre total de requêtes HTTP adressées à un conteneur MediaStore, séparées par le type d'opération (`Put`, `Get`, `Delete`, `Describe`, `List`). Unités : nombre Dimensions valides : • Nom du conteneur • Nom du groupe d'objets • Type de demande

Métrique	Description
	Statistiques valides : somme
`4xxErrorCount`	Le nombre de requêtes HTTP MediaStore qui lui ont été adressées a généré une erreur 4xx. Unités : nombre Dimensions valides : • Nom du conteneur • Nom du groupe d'objets • Type de demande Statistiques valides : somme
`5xxErrorCount`	Le nombre de requêtes HTTP MediaStore qui lui ont été adressées a généré une erreur 5xx. Unités : nombre Dimensions valides : • Nom du conteneur • Nom du groupe d'objets • Type de demande Statistiques valides : somme

Métrique	Description
`BytesUploaded`	Le nombre d'octets téléchargés pour les demandes adressées à un conteneur MediaStore , qui incluent un corps de texte. Unités : octets Dimensions valides : • Nom du conteneur • Nom du groupe d'objets Statistiques valides : Moyenne (octets par demande), Somme (octets par période), Nombre d'échantillons, Min (identique à P0.0), Max (identique à p100), tout percentile compris entre p0.0 et p99.9
`BytesDownloaded`	Le nombre d'octets téléchargés pour les demandes adressées à un compartiment MediaStore , et dont la réponse inclut un corps de texte. Unités : octets Dimensions valides : • Nom du conteneur • Nom du groupe d'objets Statistiques valides : Moyenne (octets par demande), Somme (octets par période), Nombre d'échantillons, Min (identique à P0.0), Max (identique à p100), tout percentile compris entre p0.0 et p99.9

Métrique	Description
`TotalTime`	Le nombre de millisecondes (ms) pendant lesquelles la demande était en cours du point de vue du serveur. Cette valeur est mesurée à partir du moment où MediaStore vous recevez votre demande jusqu'au moment où elle envoie le dernier octet de la réponse. Cette valeur est mesurée depuis la perspective du serveur, car les mesures effectuées depuis la perspective du client sont affectées par la latence du réseau. Unités : millisecondes Dimensions valides : • Nom du conteneur • Nom du groupe d'objets • Type de demande Statistiques valides : Moyenne, Min (identique à P0.0), Max (identique à p100), tout percentile compris entre p0.0 et p100
`TurnaroundTime`	Le nombre de millisecondes MediaStore passées à traiter votre demande. Cette valeur est mesurée à partir du moment où MediaStore vous recevez le dernier octet de votre demande jusqu'au moment où elle envoie le premier octet de la réponse. Unités : millisecondes Dimensions valides : • Nom du conteneur • Nom du groupe d'objets • Type de demande Statistiques valides : Moyenne, Min (identique à P0.0), Max (identique à p100), tout percentile compris entre p0.0 et p100

Métrique	Description
`ThrottleCount`	Le nombre de requêtes HTTP MediaStore qui lui ont été adressées a été limité. Unités : nombre Dimensions valides : • Nom du conteneur • Nom du groupe d'objets • Type de demande Statistiques valides : somme

Balisage des ressources AWS Elemental MediaStore

Une balise est une étiquette d'attribut personnalisée que vous attribuez ou attribuez à une AWS ressource. AWS Chaque balise se compose de deux parties :

- Une clé de balise (par exemple, `CostCenter`, `Environment` ou `Project`). Les clés de balises sont sensibles à la casse.
- Un champ facultatif appelé valeur de balise (par exemple, `111122223333` ou `Production`). Si la valeur de balise est identique à l'utilisation d'une chaîne vide. Les valeurs de balise sont sensibles à la casse, tout comme les clés de balise.

Les balises vous permettent d'effectuer les actions suivantes :

- Identifiez et organisez vos AWS ressources. De nombreux services AWS prennent en charge le balisage. Vous pouvez donc attribuer la même balise à des ressources à partir de différents services pour indiquer que les ressources sont liées. Par exemple, vous pouvez attribuer la même balise à un AWS Elemental MediaStore *`container`* que celle que vous attribuez à une AWS Elemental MediaLive entrée.
- Suivre vos coûts AWS. Vous activez ces balises sur le AWS Billing and Cost Management tableau de bord. AWS utilise les balises pour classer vos coûts par catégorie et vous fournir un rapport

de répartition des coûts mensuels. Pour plus d'informations, consultez Utilisation des balises d'allocation des coûts dans le Guide de l'utilisateur AWS Billing.

Les sections suivantes fournissent plus d'informations sur les balises pour AWS Elemental MediaStore.

Ressources prises en charge dans AWS Elemental MediaStore

Les ressources suivantes disponibles dans AWS Elemental MediaStore prennent en charge le balisage :

- *container*

Pour obtenir des informations sur l'ajout et la gestion de balises, veuillez consulter Gestion des tags.

AWS Elemental MediaStore ne prend pas en charge la fonctionnalité de contrôle d'accès basée sur les balises de Gestion des identités et des accès AWS (IAM).

Conventions de dénomination et d'utilisation des balises

Les conventions de dénomination et d'utilisation de base suivantes s'appliquent à l'utilisation de balises avec les ressources AWS Elemental MediaStore :

- Chaque ressource peut avoir un maximum de 50 balises.
- Pour chaque ressource, chaque clé de balise doit être unique, et chaque clé de balise peut avoir une seule valeur.
- La longueur maximale des clés de balise est de 128 caractères Unicode en UTF-8.
- La longueur maximale des valeurs de balise est de 256 caractères Unicode en UTF-8.
- Les caractères autorisés sont les lettres, les espaces et les chiffres représentables en UTF-8, ainsi que les caractères spéciaux suivants : . : + = @ _ / - (tiret). Les EC2 ressources Amazon autorisent tous les personnages.
- Les clés et les valeurs des balises distinguent les majuscules et minuscules. La bonne pratique consiste à choisir une politique pour mettre des balises en majuscule et mettre en œuvre cette politique de manière cohérente sur tous les types de ressources. Par exemple, décidez si vous souhaitez utiliser `Costcenter`, `costcenter` ou `CostCenter`, et utilisez la même convention pour toutes les balises. Évitez d'utiliser des balises avec une incohérence de traitement de cas similaires.

- Le `aws:` préfixe est interdit pour les balises ; il est réservé à l' AWS usage. Vous ne pouvez pas modifier ni supprimer des clés ou valeurs d'étiquette ayant ce préfixe. Les balises avec ce préfixe ne sont pas prises en comptes dans vos balises pour le quota de ressources.

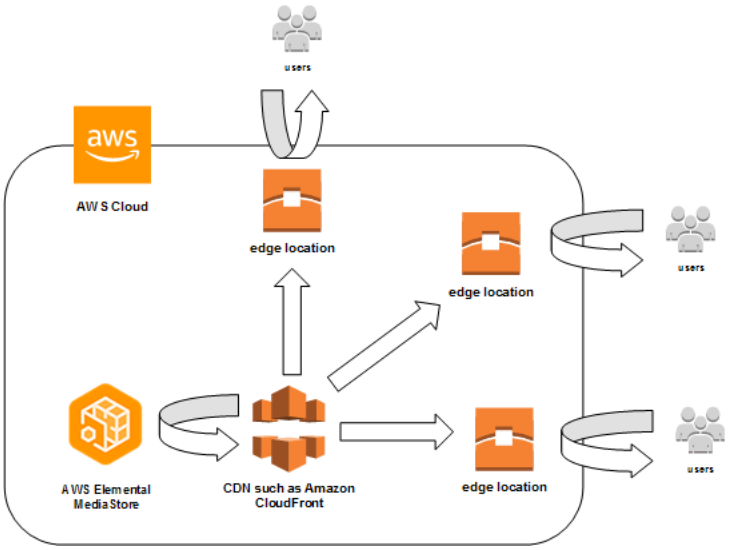
Gestion des tags

Les balises sont constituées des propriétés `Key` et `Value` sur une ressource. Vous pouvez utiliser l'API AWS CLI ou l' MediaStore API pour ajouter, modifier ou supprimer les valeurs de ces propriétés. Pour plus d'informations sur l'utilisation des balises, consultez les sections suivantes du manuel AWS Elemental MediaStore API Reference :

- CreateContainer
- ListTagsForResource
- Ressources
- TagResource
- UntagResource

Utilisation de réseaux de diffusion de contenu (CDNs)

Vous pouvez utiliser un réseau de diffusion de contenu (CDN) tel qu'Amazon CloudFront pour diffuser le contenu que vous stockez dans AWS MediaStore Elemental. Un CDN est un ensemble de serveurs réparti à l'international qui met en cache du contenu tel que des vidéos. Lorsqu'un utilisateur demande votre contenu, le CDN achemine la demande jusqu'à l'emplacement périphérique qui fournit la plus faible latence. Si votre contenu est déjà mis en cache dans cet emplacement périphérique, le CDN le diffuse immédiatement. Si votre contenu ne se trouve pas actuellement dans cet emplacement périphérique, le CDN le récupère depuis votre point d'origine (tel que votre MediaStore conteneur) et le distribue à l'utilisateur.



Rubriques

- Autoriser Amazon CloudFront à accéder à votre conteneur AWS Elemental MediaStore
- Interaction d'AWS Elemental avec MediaStore les caches HTTP

Autoriser Amazon CloudFront à accéder à votre conteneur AWS Elemental MediaStore

Vous pouvez utiliser Amazon CloudFront pour diffuser le contenu que vous stockez dans un conteneur dans AWS Elemental MediaStore. Vous pouvez le faire de l'une des manières suivantes :

- Utilisation du contrôle d'accès d'origine (OAC)- (Recommandé) Utilisez cette option si vous êtes Région AWS compatible avec la fonctionnalité OAC de CloudFront.

- Utilisation de secrets partagés- Utilisez cette option si vous n'êtes Région AWS pas compatible avec la fonctionnalité OAC de CloudFront.

Utilisation du contrôle d'accès d'origine (OAC)

Vous pouvez utiliser la fonctionnalité Origin Access Control (OAC) d'Amazon CloudFront pour sécuriser les origines AWS MediaStore Elemental avec une sécurité améliorée. Vous pouvez activer AWS la version 4 de signature (SigV4) sur les CloudFront demandes d' MediaStoreorigine et définir quand et si vous CloudFront devez signer les demandes. Vous pouvez accéder à la fonctionnalité OAC CloudFront par le biais de la console APIs, du SDK ou de la CLI, sans frais supplémentaires pour son utilisation.

Pour plus d'informations sur l'utilisation de la fonctionnalité OAC avec MediaStore, consultez Restreindre l'accès à une MediaStore origine dans le manuel Amazon CloudFront Developer Guide.

Utilisation de secrets partagés

Si vous Région AWS n'êtes pas compatible avec la fonctionnalité OAC d'Amazon CloudFront, vous pouvez associer à votre conteneur AWS MediaStore Elemental une politique qui accorde un accès en lecture ou supérieur à. CloudFront

> **Note**
>
> Nous vous recommandons d'utiliser la fonctionnalité OAC si vous la Région AWS supportez. Les procédures suivantes nécessitent que vous configuriez MediaStore et CloudFront que vous utilisiez des secrets partagés afin de restreindre l'accès aux MediaStore conteneurs. Pour respecter les meilleures pratiques de sécurité, cette configuration manuelle nécessite une rotation périodique des secrets. Avec OAC on MediaStore origin, vous pouvez demander de signer les demandes CloudFront à l'aide de SigV4 et de les transmettre à des MediaStore fins de correspondance des signatures, éliminant ainsi le besoin d'utiliser et de faire pivoter les secrets. Cela garantit que les demandes sont automatiquement vérifiées avant que le contenu multimédia ne soit diffusé, ce qui CloudFront simplifie MediaStore et sécurise la diffusion du contenu multimédia.

Pour autoriser CloudFront l'accès à votre conteneur (console)

1. Ouvrez la MediaStore console à l'adresse https://console.aws.amazon.com/mediastore/.

2. Sur la page Containers (Conteneurs), choisissez le nom du conteneur.

   La page des détails du conteneur s'affiche.

3. Dans la section Politique relative aux conteneurs, joignez une politique qui accorde un accès en lecture ou supérieur à Amazon CloudFront.

   Example

   L'exemple de politique suivant, qui est similaire à l'exemple de politique pour l'accès public en lecture via HTTPS, répond à ces exigences car il autorise `GetObject` et `DescribeObject` commande toute personne soumettant des demandes à votre domaine via HTTPS. En outre, l'exemple de politique suivant sécurise mieux votre flux de travail car il autorise CloudFront l'accès aux MediaStore objets uniquement lorsque la demande est effectuée via une connexion HTTPS et contient l'en-tête Referer correct.

4. Dans la section Container CORS policy (Stratégie CORS du conteneur), attribuez une stratégie qui accorde le niveau d'accès approprié.

   > **Note**
   >
   > Une stratégie CORS est nécessaire uniquement si vous souhaitez fournir un accès à un joueur basé sur navigateur.

5. Notez les détails suivants :

   - Le point de terminaison des données attribué à votre conteneur . Vous pouvez trouver ces informations dans la section Info (Infos) de la page Containers (Conteneurs). Dans CloudFront, le point de terminaison des données est appelé nom de domaine d'origine.
   - La structure de dossiers du conteneur dans lequel les objets sont stockés. Dans CloudFront, c'est ce que l'on appelle le chemin d'origine. Notez que ce paramètre est facultatif. Pour plus d'informations sur les chemins d'origine, consultez le manuel Amazon CloudFront Developer Guide.

6. Dans CloudFront, créez une distribution configurée pour diffuser le contenu d'AWS Elemental MediaStore. Vous aurez besoin des informations collectées dans l'étape précédente.

Après avoir attaché la politique à vos MediaStore conteneurs, vous devez configurer CloudFront pour n'utiliser que des connexions HTTPS pour les demandes d'origine, et également ajouter un en-tête personnalisé avec la valeur secrète correcte.

Pour configurer CloudFront l'accès à votre conteneur via une connexion HTTPS avec une valeur secrète pour l'en-tête Referer (console)

1. Ouvrez la CloudFront console.
2. Sur la page Origines, choisissez votre MediaStore origine.
3. Choisissez Modifier.
4. Choisissez HTTPS uniquement pour le protocole.
5. Dans la section Ajouter un en-tête personnalisé, choisissez Ajouter un en-tête.
6. Pour le nom, choisissez Referer. Pour la valeur, utilisez la même *<secretValue>* chaîne que celle que vous avez utilisée dans votre politique de conteneur.
7. Choisissez Enregistrer et laissez les modifications se déployer.

Interaction d'AWS Elemental avec MediaStore les caches HTTP

AWS Elemental MediaStore stocke les objets afin qu'ils puissent être mis en cache correctement et efficacement par les réseaux de diffusion de contenu () CDNs tels qu'Amazon. CloudFront Lorsqu'un utilisateur final ou un CDN récupère un objet MediaStore, le service renvoie des en-têtes HTTP qui affectent le comportement de mise en cache de l'objet. (Les normes relatives au comportement de mise en cache du protocole HTTP 1.1 se trouvent dans RFC2616 la section 13.) Ces en-têtes sont :

- **`ETag`** (non personnalisable) : l'en-tête de balise d'entité est un identificateur unique pour la réponse envoyée par MediaStore . Les navigateurs Web CDNs et conformes aux normes utilisent cette balise comme clé pour mettre en cache l'objet. MediaStore génère automatiquement un `ETag` pour chaque objet lorsqu'il est chargé. Vous pouvez consulter les détails d'un objet pour déterminer sa ETag valeur.
- **`Last-Modified`**(non personnalisable) — La valeur de cet en-tête indique la date et l'heure de modification de l'objet. MediaStore génère automatiquement cette valeur lorsque l'objet est chargé.
- **`Cache-Control`** (personnalisable) – La valeur de cet en-tête contrôle la durée pendant laquelle un objet doit être mis en cache avant que le CDN vérifie s'il a été modifié. Vous pouvez attribuer à cet en-tête n'importe quelle valeur lorsque vous téléchargez un objet dans un MediaStore conteneur à l'aide de la CLI ou de l'API. L'ensemble complet de valeurs valides est décrit dans la documentation HTTP/1.1. Si vous ne définissez pas cette valeur lorsque vous chargez un objet, cet en-tête MediaStore ne sera pas renvoyé lors de la récupération de l'objet.

L'en-tête Cache-Control est souvent utilisé pour spécifier une durée de mise en cache de l'objet. Par exemple, supposons que vous ayez un fichier manifeste vidéo qui est fréquemment écrasé par un encodeur. Vous pouvez définir `max-age` sur 10 pour indiquer que l'objet doit être mis en cache pendant seulement 10 secondes. Ou supposons que vous ayez un segment vidéo stocké qui ne sera jamais remplacé. Vous pouvez définir le `max-age` pour cet objet sur 31536000 à mettre en cache pendant environ 1 an.

Demandes conditionnelles

Demandes conditionnelles adressées à MediaStore

MediaStore répond de manière identique aux demandes conditionnelles (en utilisant des en-têtes de demande tels que `If-Modified-Since` et`If-None-Match`, comme décrit dans RFC7232) et aux demandes inconditionnelles. Cela signifie que lorsqu'il MediaStore reçoit une `GetObject` demande valide, le service renvoie toujours l'objet même si le client possède déjà l'objet.

Demandes conditionnelles adressées à CDNs

CDNs qui diffusent du contenu pour le compte de MediaStore peuvent traiter les demandes conditionnelles en les renvoyant`304 Not Modified`, comme décrit dans RFC7232 la section 4.1. Cela indique qu'il n'est pas nécessaire de transférer le contenu complet de l'objet, car le demandeur possède déjà un objet qui correspond à la demande conditionnelle.

CDNs (et d'autres caches conformes à HTTP/1.1) basent ces décisions sur les `Cache-Control` en-têtes `ETag` et transmis par les serveurs d'origine. Pour contrôler la fréquence CDNs à laquelle les serveurs MediaStore d'origine demandent des mises à jour à des objets récupérés à plusieurs reprises, définissez `Cache-Control` les en-têtes de ces objets lorsque vous les MediaStore téléchargez.

Utilisation de ce service avec un AWS SDK

AWS des kits de développement logiciel (SDKs) sont disponibles pour de nombreux langages de programmation courants. Chaque SDK fournit une API, des exemples de code et de la documentation qui facilitent la création d'applications par les développeurs dans leur langage préféré.

Documentation SDK	Exemples de code
AWS SDK pour C++	AWS SDK pour C++ exemples de code
AWS CLI	AWS CLI exemples de code
AWS SDK pour Go	AWS SDK pour Go exemples de code
AWS SDK pour Java	AWS SDK pour Java exemples de code
AWS SDK pour JavaScript	AWS SDK pour JavaScript exemples de code
AWS SDK pour Kotlin	AWS SDK pour Kotlin exemples de code
AWS SDK pour .NET	AWS SDK pour .NET exemples de code
AWS SDK pour PHP	AWS SDK pour PHP exemples de code
Outils AWS pour PowerShell	Outils AWS pour PowerShell exemples de code
AWS SDK pour Python (Boto3)	AWS SDK pour Python (Boto3) exemples de code
AWS SDK pour Ruby	AWS SDK pour Ruby exemples de code
AWS SDK pour Rust	AWS SDK pour Rust exemples de code
AWS SDK pour SAP ABAP	AWS SDK pour SAP ABAP exemples de code
AWS SDK pour Swift	AWS SDK pour Swift exemples de code

Pour voir des exemples spécifiques à ce service, consultez Exemples de code pour MediaStore l'utilisation AWS SDKs.

Exemple de disponibilité

Vous n'avez pas trouvé ce dont vous avez besoin ? Demandez un exemple de code en utilisant le lien Provide feedback (Fournir un commentaire) en bas de cette page.

Exemples de code pour MediaStore l'utilisation AWS SDKs

Les exemples de code suivants montrent comment utiliser MediaStore un kit de développement AWS logiciel (SDK).

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit de développement logiciel (SDK).

Exemples de code

- Exemples de base pour MediaStore l'utilisation AWS SDKs
  - Actions d' MediaStore utilisation AWS SDKs
    - Utilisation CreateContainer avec un AWS SDK ou une CLI
    - Utilisation DeleteContainer avec un AWS SDK ou une CLI
    - Utilisation DeleteObject avec un AWS SDK
    - Utilisation DescribeContainer avec un AWS SDK ou une CLI
    - Utilisation GetObject avec un AWS SDK ou une CLI
    - Utilisation ListContainers avec un AWS SDK ou une CLI
    - Utilisation PutObject avec un AWS SDK ou une CLI

Exemples de base pour MediaStore l'utilisation AWS SDKs

Les exemples de code suivants montrent comment utiliser les principes de base de AWS Elemental MediaStore with AWS SDKs.

Exemples

- Actions d' MediaStore utilisation AWS SDKs
  - Utilisation CreateContainer avec un AWS SDK ou une CLI
  - Utilisation DeleteContainer avec un AWS SDK ou une CLI

- Utilisation DeleteObject avec un AWS SDK
- Utilisation DescribeContainer avec un AWS SDK ou une CLI
- Utilisation GetObject avec un AWS SDK ou une CLI
- Utilisation ListContainers avec un AWS SDK ou une CLI
- Utilisation PutObject avec un AWS SDK ou une CLI

Actions d' MediaStore utilisation AWS SDKs

Les exemples de code suivants montrent comment effectuer des MediaStore actions individuelles avec AWS SDKs. Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions pour configurer et exécuter le code.

Les exemples suivants incluent uniquement les actions les plus couramment utilisées. Pour obtenir la liste complète, veuillez consulter la AWS Elemental MediaStore Référence d'API.

Exemples

- Utilisation CreateContainer avec un AWS SDK ou une CLI
- Utilisation DeleteContainer avec un AWS SDK ou une CLI
- Utilisation DeleteObject avec un AWS SDK
- Utilisation DescribeContainer avec un AWS SDK ou une CLI
- Utilisation GetObject avec un AWS SDK ou une CLI
- Utilisation ListContainers avec un AWS SDK ou une CLI
- Utilisation PutObject avec un AWS SDK ou une CLI

Utilisation **`CreateContainer`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `CreateContainer`.

CLI

AWS CLI

Pour créer un conteneur

L'`create-container`exemple suivant crée un nouveau conteneur vide.

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

Sortie :

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

Pour plus d'informations, consultez la section Création d'un conteneur dans le guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous CreateContainerà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

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

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

```
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateContainer {
    public static long sleepTime = 10;

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

                Usage:    <containerName>

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

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

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

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


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

            CreateContainerResponse containerResponse =
 mediaStoreClient.createContainer(containerRequest);
            String status = containerResponse.container().status().toString();
```

```
            while (!status.equalsIgnoreCase("Active")) {
                status = DescribeContainer.checkContainer(mediaStoreClient,
 containerName);
                System.out.println("Status - " + status);
                Thread.sleep(sleepTime * 1000);
            }

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

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

- Pour plus de détails sur l'API, reportez-vous CreateContainerà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteContainer`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DeleteContainer`.

CLI

AWS CLI

Pour supprimer un conteneur

L'`delete-container`exemple suivant supprime le conteneur spécifié. Vous pouvez supprimer un conteneur uniquement s'il ne possède aucun objet.

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

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression d'un conteneur dans le guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous DeleteContainerà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

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

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

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

                Usage:    <containerName>

                Where:
```

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

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

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

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


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

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

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

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

```
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteContainerà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteObject`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`DeleteObject`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

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

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

```
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteObject {
    public static void main(String[] args) throws URISyntaxException {
        final String usage = """

                Usage:    <completePath> <containerName>

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

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

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

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

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

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

```
            mediaStoreData.deleteObject(deleteObjectRequest);

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

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

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

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

- Pour plus de détails sur l'API, reportez-vous DeleteObjectà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DescribeContainer`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DescribeContainer`.

CLI

AWS CLI

Pour afficher les détails d'un conteneur

L'`describe-container`exemple suivant affiche les détails du conteneur spécifié.

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

Sortie :

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

Pour plus d'informations, consultez la section Affichage des détails d'un conteneur dans le Guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous DescribeContainerà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

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

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

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

                Usage:    <containerName>

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

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

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

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

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

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

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

- Pour plus de détails sur l'API, reportez-vous DescribeContainerà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetObject`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `GetObject`.

CLI

AWS CLI

Pour télécharger un objet

L'`get-object`exemple suivant télécharge un objet vers le point de terminaison spécifié.

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

Sortie :

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

Pour télécharger une partie d'un objet

L'`get-object`exemple suivant télécharge une partie d'un objet vers le point de terminaison spécifié.

```
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
```

Sortie :

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

Pour plus d'informations, consultez la section Téléchargement d'un objet dans le guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous GetObjectà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

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

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

                Usage:    <completePath> <containerName> <savePath>
```

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

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

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

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

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

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

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

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

            File targetFile = new File(savePath);
```

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

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

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

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

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

- Pour plus de détails sur l'API, reportez-vous GetObjectà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`ListContainers`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `ListContainers`.

CLI

AWS CLI

Pour consulter la liste des conteneurs

L'`list-containers`exemple suivant affiche la liste de tous les conteneurs associés à votre compte.

```
aws mediastore list-containers
```

Sortie :

```
{
    "Containers": [
        {
            "CreationTime": 1505317931,
            "Endpoint": "https://aaabbbcccdddee.data.mediastore.us-
west-2.amazonaws.com",
            "Status": "ACTIVE",
            "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleLiveDemo",
            "AccessLoggingEnabled": false,
            "Name": "ExampleLiveDemo"
        },
        {
            "CreationTime": 1506528818,
            "Endpoint": "https://fffggghhhiiijj.data.mediastore.us-
west-2.amazonaws.com",
            "Status": "ACTIVE",
            "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer",
            "AccessLoggingEnabled": false,
            "Name": "ExampleContainer"
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage d'une liste de conteneurs dans le guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous ListContainersà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

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

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

    public static void main(String[] args) {

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

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

    public static void listAllContainers(MediaStoreClient mediaStoreClient) {
        try {
```

```
            ListContainersResponse containersResponse =
 mediaStoreClient.listContainers();
            List<Container> containers = containersResponse.containers();
            for (Container container : containers) {
                System.out.println("Container name is " + container.name());
            }

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

- Pour plus de détails sur l'API, reportez-vous ListContainersà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutObject`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `PutObject`.

CLI

AWS CLI

Pour télécharger un objet

L'`put-object`exemple suivant télécharge un objet dans le conteneur spécifié. Vous pouvez spécifier le chemin du dossier dans lequel l'objet sera enregistré dans le conteneur. Si le dossier existe déjà, AWS Elemental y MediaStore stocke l'objet. Si le dossier n'existe pas, le service le crée, puis y stocke l'objet.

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

```
    --content-type binary/octet-stream
```

Sortie :

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

Pour plus d'informations, consultez la section Chargement d'un objet dans le guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous PutObjectà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

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

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

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

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

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

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

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

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

    public static void putMediaObject(MediaStoreDataClient mediaStoreData, String
 filePath, String completePath) {
        try {
            File myFile = new File(filePath);
```

```
            RequestBody requestBody = RequestBody.fromFile(myFile);

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

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

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

    public static String getEndpoint(String containerName) {

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

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

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

- Pour plus de détails sur l'API, reportez-vous PutObjectà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Quotas dans AWS Elemental MediaStore

La console Service Quotas fournit des informations sur les quotas AWS Elemental MediaStore . En plus de visualiser les quotas par défaut, vous pouvez utiliser la console Quotas de service pour demander des augmentations de quotas pour les quotas ajustables.

Le tableau suivant décrit les quotas, anciennement appelés limites, dans AWS Elemental MediaStore. Les quotas sont le nombre maximal de ressources ou d'opérations de service pour votre compte AWS.

> **Note**
>
> Pour attribuer des quotas à des conteneurs individuels au sein de votre compte, contactez AWS Support ou votre responsable de compte. Cette option peut vous aider à répartir les limites au niveau du compte entre vos conteneurs, afin d'éviter qu'un conteneur n'utilise la totalité de votre quota.

Ressource ou opération	Quota par défaut	Commentaires
Conteneurs	100	Nombre maximum de conteneurs que vous pouvez créer dans ce compte.
Niveaux du dossier	10	Nombre maximum de niveaux de dossier que vous pouvez créer dans un conteneur. Vous pouvez créer autant de dossiers que vous le souhaitez, à condition qu'ils ne soient pas imbriqués sur plus de 10 niveaux au sein d'un conteneur.
Dossiers	Illimité	Vous pouvez créer autant de dossiers que vous le souhaitez, à condition qu'ils ne soient pas imbriqués sur plus de 10 niveaux au sein d'un conteneur.
Taille de l'objet	25 Mo	Taille maximum d'un seul objet.

Ressource ou opération	Quota par défaut	Commentaires
Objets	Illimité	Vous pouvez charger autant d'objets que vous le souhaitez dans un dossier ou un conteneur de votre compte.
Taux des demandes d'API DeleteObject	100	Nombre maximum de demandes d'opérations par seconde. Les autres demandes sont bloquées. Vous pouvez demander une augmentation de quota.
Taux des demandes d'API DescribeObject	1 000	Nombre maximum de demandes d'opérations par seconde. Les autres demandes sont bloquées. Vous pouvez demander une augmentation de quota.
Taux de demandes d'GetObjectAPI pour la disponibilité des téléchargements standard	1 000	Nombre maximum de demandes d'opérations par seconde. Les autres demandes sont bloquées. Vous pouvez demander une augmentation de quota.
Taux de demandes d'GetObject API pour la disponibilité des téléchargements en streaming	25	Nombre maximum de demandes d'opérations par seconde. Les autres demandes sont bloquées. Vous pouvez demander une augmentation de quota.
Taux des demandes d'API ListItems	5	Nombre maximum de demandes d'opérations par seconde. Les autres demandes sont bloquées. Vous pouvez demander une augmentation de quota.

Ressource ou opération	Quota par défaut	Commentaires
Taux de demandes d'PutObjectAPI pour le codage de transfert fragmenté (également connu sous le nom de disponibilité du téléchargement en continu)	10	Nombre maximum de demandes d'opérations par seconde. Les autres demandes sont bloquées. Vous pouvez demander une augmentation de quota. Dans la demande, spécifiez le TPS demandé et la taille moyenne de l'objet.
Taux de demandes d'PutObjectAPI pour la disponibilité des téléchargements standard	100	Nombre maximum de demandes d'opérations par seconde. Les autres demandes sont bloquées. Vous pouvez demander une augmentation de quota. Dans la demande, spécifiez le TPS demandé et la taille moyenne de l'objet.
Règles d'une stratégie de métriques	10	Le nombre maximal de règles que vous pouvez inclure dans une stratégie de métriques.
Règles dans une stratégie de cycle de vie des objets	10	Nombre maximal de règles que vous pouvez inclure dans une stratégie de cycle de vie des objets.

Informations relatives à AWS Elemental MediaStore

Le tableau suivant répertorie les ressources connexes qui vous seront utiles lorsque vous travaillerez avec AWS Elemental MediaStore.

- Cours et ateliers — Liens vers des cours spécialisés et basés sur des rôles, ainsi que des ateliers à votre rythme pour vous aider à perfectionner vos AWS compétences et à acquérir une expérience pratique.
- AWS Centre pour développeurs : découvrez les didacticiels, téléchargez des outils et découvrez les événements AWS destinés aux développeurs.
- AWS Outils de développement : liens vers des outils de développement SDKs, des boîtes à outils IDE et des outils de ligne de commande pour le développement et la gestion d' AWS applications.
- Centre de ressources pour la mise en route : découvrez comment configurer votre application Compte AWS, rejoindre la AWS communauté et lancer votre première application.
- Tutoriels pratiques — Suivez les step-by-step didacticiels pour lancer votre première application sur AWS.
- AWS Livres blancs : liens vers une liste complète de livres AWS blancs techniques, traitant de sujets tels que l'architecture, la sécurité et l'économie, rédigés par des architectes de AWS solutions ou d'autres experts techniques.
- AWS Support Centre — Le centre de création et de gestion de vos AWS Support dossiers. Comprend également des liens vers d'autres ressources utiles, telles que des forums, des informations techniques FAQs, l'état de santé des services et AWS Trusted Advisor.
- Support— La principale page Web contenant des informations sur Support un one-on-one canal d'assistance à réponse rapide pour vous aider à créer et à exécuter des applications dans le cloud.
- Contactez-nous : point de contact central pour toute question relative à la facturation AWS , à votre compte, aux événements, à des abus ou à d'autres problèmes.
- AWS Conditions du site — Informations détaillées sur nos droits d'auteur et notre marque commerciale ; votre compte, votre licence et l'accès au site ; et d'autres sujets.

Historique du document pour le guide de l'utilisateur

Le tableau suivant décrit la documentation de cette version d'AWS Elemental MediaStore. Pour recevoir les notifications de mise à jour de cette documentation, abonnez-vous à un flux RSS.

Modification	Description	Date
Avis de fin de support	Avis de fin de support : le 13 novembre 2025, le support d'AWS MediaStore Elemental AWS sera interrompu. Après le 13 novembre 2025, vous ne pourrez plus accéder à la MediaStore console ni aux MediaStore ressources. Pour plus d'informations, consultez ce billet de blog.	12 novembre 2024
Amélioration du contrôle d'accès à l'origine (OAC)	Ajout d'informations sur l'utilisation de l'OAC avec AWS MediaStore Elemental.	17 avril 2023
Mise à jour des quotas	Valeur du quota et description corrigées pour`Rules in a Metric Policy`.	25 octobre 2022
ExpiresAt champ	Les journaux d'accès incluent désormais un `ExpiresAt` champ qui indique la date et l'heure d'expiration de l'objet en fonction des règles relatives aux données transitoires définies dans la politique de cycle de vie du conteneur.	16 juillet 2020
Règles de transition du cycle de vie	Vous pouvez désormais ajouter une règle de transitio	20 avril 2020

	n du cycle de vie à votre stratégie de cycle de vie des objets qui définit les objets à déplacer vers la classe de stockage IA (accès peu fréquent) une fois qu'ils ont atteint un certain âge.	
Récipient vide	Vous pouvez désormais supprimer tous les objets d'un conteneur simultanément.	7 avril 2020
Support pour les CloudWatch métriques Amazon	Vous pouvez définir une politique de mesures pour dicter à quelles métriques sont MediaStore envoyées CloudWatch.	30 mars 2020
Caractères génériques dans les règles de suppression d'objets	Dans une stratégie de cycle de vie d'objet, vous pouvez désormais utiliser un caractère générique dans une règle de suppression d'objet. Cela vous permet de spécifier, en fonction de leur nom de fichier ou de leur extension, les fichiers que vous souhaitez que le service supprime après un certain nombre de jours.	20 décembre 2019
Règles de cycle de vie des objets	Vous pouvez désormais ajouter une règle à votre stratégie de cycle de vie des objets qui indique une expiration par âge en secondes.	13 septembre 2019

CloudFormation soutien	Vous pouvez désormais utiliser un CloudFormation modèle pour créer un conteneur automatiquement. Le modèle CloudFormation gère les données pour les cinq actions d'API : création d'un conteneur, définition de la journalisation des accès, mise à jour de la stratégie de conteneur par défaut, ajout d'une stratégie de partage des ressources cross-origin (CORS), et en ajoutant une stratégie de cycle de vie des objets.	17 mai 2019
Quotas de disponibilité des téléchargements en streaming	Pour les objets avec une disponibilité de chargement en streaming (transfert fragmenté des objets), l'opération `PutObject` ne peut pas dépasser 10 TPS et l'opération `GetObject` ne peut pas dépasser 25 TPS.	8 avril 2019
Transfert fragmenté d'objets	Ajout de la prise en charge du transfert fragmenté des objets. Cette fonctionnalité vous permet de spécifier qu'un objet est disponible pour le téléchargement avant que l'objet soit complètement chargé.	5 avril 2019

Journalisation des accès	AWS Elemental prend MediaStore désormais en charge la journalisation des accès, qui fournit des enregistrements détaillés des demandes adressées aux objets d'un conteneur.	25 février 2019
Règles de cycle de vie des objets	Ajout de la prise en charge de stratégies de cycle de vie des objets qui régissent la date d'expiration des objets dans le conteneur actuel.	12 décembre 2018
Augmentation du quota de taille des objets	Le quota pour la taille d'un objet est désormais de 25 Mo.	10 octobre 2018
Augmentation du quota de taille des objets	Le quota pour la taille d'un objet est désormais de 20 Mo.	6 septembre 2018
AWS CloudTrail intégration	Le contenu de CloudTrai l l'intégration a été mis à jour pour tenir compte des modifications récentes apportées au CloudTrail service.	12 juillet 2018
Collaboration avec le CDN	Ajout d'informations sur l'utilisation d'AWS Elemental MediaStore avec un réseau de diffusion de contenu (CDN) tel qu'Amazon. CloudFront	14 avril 2018

Configurations CORS	AWS Elemental prend MediaStore désormais en charge le partage de ressources entre origines (CORS), qui permet aux applications Web clientes chargées dans un domaine d'interagir avec les ressources d'un autre domaine.	7 février 2018
Nouveau guide et service	Il s'agit de la version initiale du service de création et de stockage de vidéos, AWS MediaStore Elemental, et du guide de l'utilisateur AWS MediaStore Elemental.	27 novembre 2017

Note

- Les services AWS multimédias ne sont pas conçus ou destinés à être utilisés avec des applications ou dans des situations nécessitant des performances fiables, telles que les opérations de sécurité des personnes, les systèmes de navigation ou de communication, le contrôle du trafic aérien ou les appareils de survie dans lesquels l'indisponibilité, l'interruption ou la défaillance des services pourraient entraîner la mort, des blessures, des dommages matériels ou des dommages environnementaux.

AWS Glossaire

Pour la AWS terminologie la plus récente, consultez le AWS glossaire dans la Glossaire AWS référence.